



**TUGAS AKHIR - TE 141599**

**PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE  
PID - *LINEAR QUADRATIC REGULATOR***

Fanniesha Hamada  
NRP 2211100207

Dosen Pembimbing  
Ir. Rusdhianto Effendie A.K., MT.  
Ir. Ali Fatoni, MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**TUGAS AKHIR - TE 141599**

**PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE  
PID - *LINEAR QUADRATIC REGULATOR***

Fanniesha Hamada  
NRP 2211100207

Dosen Pembimbing  
Ir. Rusdhianto Effendie A.K., MT.  
Ir. Ali Fatoni, MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015





**FINAL PROJECT - TE 141599**

**VELOCITY CONTROL OF PARALLEL HYBRID ELECTRIC  
VEHICLE SIMULATOR USING PID - LINEAR QUADRATIC  
REGULATOR METHOD**

Fanniesha Hamada  
NRP 2211100207

Advisor  
Ir. Rusdhianto Effendie A.K., MT.  
Ir. Ali Fatoni, MT.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

**PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL*  
*HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE  
PID - *LINEAR QUADRATIC REGULATOR***

**TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

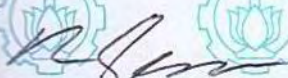
**Pada**

**Bidang Studi Teknik Sistem Pengaturan  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

  
**Ir. Rusdhianto Effendie A.K., MT.**  
**NIP. 195704241985021001**

  
**Ir. Ali Fatoni, MT.**  
**NIP. 196206031989031002**



**PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE  
PID - *LINEAR QUADRATIC REGULATOR***

Fanniesha Hamada – 2211100207

Pembimbing I : Ir. Rushdianto Effendie A.K., MT.

Pembimbing II : Ir. Ali Fatoni MT.

**ABSTRAK**

*Hybrid Electric Vehicle* (HEV) merupakan suatu kendaraan dengan konsep ramah lingkungan dan hemat energi yang diharapkan menjadi salah satu alternatif menanggulangi efek rumah kaca dan krisis energi. HEV menggabungkan kinerja *Internal Combustion Engine* (ICE) atau mesin bakar dan motor listrik. Pada HEV dengan konfigurasi paralel, ICE dan motor listrik dapat bekerja bersama-sama. Pada Tugas Akhir ini digunakan Simulator *Parallel Hybrid Electric Vehicle* (PHEV). Simulator ini merepresentasikan kondisi nyata HEV namun dalam skala yang lebih kecil. Simulator ini terdiri dari mesin bakar 2 tak sebagai penggerak utama, motor DC sebagai penggerak pembantu, dan beban berupa rem magnetik arus *eddy*. Ketika terjadi permasalahan regulator akibat pembebanan lebih pada kendaraan, kecepatan putar pada ICE menurun sehingga kecepatan HEV tidak sesuai dengan *output* yang diharapkan. Oleh karena itu, dibutuhkan suatu kontroler untuk melakukan pengaturan kerja dari motor listrik agar bekerja sesuai dengan kebutuhan yang diinginkan. Kontroler PID – *Linear Quadratic Regulator* (PID-LQR) digunakan untuk memperbaiki performansi kerja HEV agar mampu membantu ICE mencapai kecepatan putar yang seharusnya. Berdasarkan hasil pengujian secara simulasi didapatkan motor DC menggunakan kontroler PID-LQR mampu membantu kinerja ICE sehingga dapat mengembalikan respon sistem menuju nilai *steady state* ketika terjadi pembebanan berlebih pada rentang arus beban rem nominal 0.56-1.14 A. Berdasarkan hasil pengujian secara implementasi, motor DC dapat membantu kinerja ICE, namun masih terdapat *error steady state*.

**Kata Kunci** : *Hybrid Electric Vehicle*, ICE, Motor Listrik, Motor DC, PID-LQR

--- Halaman ini sengaja dikosongkan ---

# **VELOCITY CONTROL OF PARALLEL HYBRID ELECTRIC VEHICLE SIMULATOR USING PID - LINEAR QUADRATIC REGULATOR METHOD**

Fanniesha Hamada – 2211100207

Advisor I : Ir. Rushdianto Effendie A.K., MT.

Advisor II : Ir. Ali Fatoni MT.

## **ABSTRACT**

*Hybrid Electric Vehicle (HEV) is a vehicle with the concept of environmentally friendly and energy saving which is expected to be an alternative to combat the greenhouse effect and energy crisis. HEV combines the performance of Internal Combustion Engine (ICE) and electric motor. In the parallel HEV configuration, ICE and electric motor can work together. Parallel Hybrid Electric Vehicle simulator is used in this final project. This simulator represents real condition of HEV but in a smaller scale. This simulator consist of a stroke-2 combustion engine as the prime mover, DC motor as an assist mover, and the load of magnetic eddy current brake. When there is excessive load on a vehicle called regulator problem, the rotational speed decreases so that the rotational speed of HEV doesn't match with the expected output. Therefore, a controller is needed for working arrangement of the electric motor according to the desired needs. PID – Linear Quadratic Regulator Controller (PID-LQR) is used to improve work performance of HEV in order to assist the combustion engine to achieve an appropriate rotating speed. Based on simulation testing result, DC motor with PID-LQR controler can help the performance of ICE so that it can restore the system response towards steady state value at excessive load in the range of 0,56-1,14 A of nominal load current brake. Based on implementation testing result, DC motor can assist the combustion engine, however there's still have an error steady state.*

**Keywords :** Hybrid Electric Vehicle, Internal Combustion Engine, Electric Motor, DC Motor, PID-LQR

*--- Halaman ini sengaja dikosongkan ---*

## **KATA PENGANTAR**

Alhamdulillah penulis ucapkan atas kehadiran Allah SWT karena atas segala rahmat dan hidayah-Nya, Tugas Akhir ini dapat terselesaikan dengan baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada bidang studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul :

### **PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE PID - *LINEAR QUADRATIC REGULATOR***

Tugas Akhir ini disusun berdasarkan pembelajaran dan pengujian yang telah dilakukan dan tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Kedua orang tua penulis yang selalu memberi dukungan penuh dalam bentuk apapun.
2. Dosen pembimbing, Bapak Ir. Rusdhianto Effendie A.K., MT. dan Ir. Ali Fatoni, MT yang selalu membimbing dan memberikan banyak ilmu dan motivasi.
3. Rekan satu tim penulis, Ajib Setiawan N., Sentosa Sondang O., Alif Ridwan M, Aulia Rahma A, dan Fahrul.
4. Angkatan e-51 yang telah memberi dukungan moril dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa buku Tugas Akhir ini belum sempurna, oleh karena itu penulis sangat mengharapkan saran dan masukan untuk perbaikan di masa yang akan datang. Semoga buku Tugas Akhir ini dapat bermanfaat dalam pengembangan ilmu dan teknologi dan juga bagi pembaca di kemudian hari.

Surabaya, 30 Juni 2015

Fanniesha Hamada  
2211100207

--- Halaman ini sengaja dikosongkan ---



## DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
HALAMAN PENGESAHAN.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan.....	3
1.6 Relevansi .....	3
BAB 2 DASAR TEORI .....	5
2.1 <i>Hybrid Electric Vehicle</i> (HEV) .....	5
2.1.1 HEV Konfigurasi Paralel .....	5
2.1.2 HEV Konfigurasi Seri .....	6
2.1.3 HEV Konfigurasi Seri-Paralel .....	7
2.2 Simulator <i>Hybrid Electric Vehicle</i> .....	9
2.2.1 <i>Internal Combustion Engine</i> (ICE) 2 Tak .....	9
2.2.2 Motor Listrik DC.....	12
2.2.3 Rem Elektromagnetik.....	16
2.2.4 <i>Timing Belt</i> .....	18
2.2.5 <i>Rotary Encoder</i> .....	22
2.3 Sistem Minimum Arduino Uno R3 .....	22
2.4 Identifikasi Sistem .....	24
2.4.1 Identifikasi Strejc .....	25
2.4.2 Identifikasi Fisik .....	27
2.5 Representasi <i>State Space</i> .....	30
2.6 <i>Root Mean Square Error</i> .....	34
2.7 Kontroler .....	34
2.7.1 Kontroler <i>Proportional Integral Derivative</i> (PID) ...	35
2.7.2 Kontroler Optimal LQR .....	38
2.7.3 Kontroler PID-LQR .....	38

BAB 3 PERANCANGAN SISTEM.....	41
3.1 Gambaran Umum Sistem .....	41
3.2 Perancangan Perangkat Keras .....	43
3.2.1 Perancangan Perangkat Mekanik .....	44
3.2.2 Perancangan Perangkat Elektronik.....	51
3.3 Perancangan Perangkat Lunak .....	55
3.3.1 <i>Software</i> Arduino .....	55
3.3.2 <i>Software</i> LabView.....	56
3.3.3 <i>Software</i> MATLAB.....	57
3.4 Pengujian Sistem.....	58
3.4.1 Pengujian Sensor.....	59
3.5 Proses Identifikasi dan Pemodelan Sistem.....	61
3.5.1 Identifikasi dan Pemodelan Motor DC.....	62
3.5.2 Identifikasi dan Pemodelan ICE .....	74
3.6 Pengujian dan Validasi.....	80
3.7 Perancangan Kontroler PID-LQR .....	82
3.7.1 Desain Simulink PID-LQR .....	85
BAB 4 PENGUJIAN DAN ANALISIS .....	87
4.1 Gambaran Umum Pengujian Sistem .....	87
4.2 Simulasi Sistem.....	87
4.2.1 Simulasi ICE Beban Nominal Tanpa Bantuan Motor Listrik.....	87
4.2.2 Simulasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID- <i>Linear Quadratic</i> <i>Regulator</i> .....	91
4.3 Implementasi Sistem .....	100
4.3.1 Implementasi ICE Beban Nominal Tanpa Bantuan Motor Listrik.....	100
4.3.2 Implementasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID- <i>Linear</i> <i>Quadratic Regulator</i> .....	102
BAB 5 PENUTUP .....	103
5.1 Kesimpulan .....	103
5.2 Saran .....	103
DAFTAR PUSTAKA .....	105
LAMPIRAN .....	107
RIWAYAT HIDUP .....	117

## DAFTAR TABEL

<b>Tabel 2.1</b>	Perbandingan Kapasitas Konfigurasi HEV .....	8
<b>Tabel 2.2</b>	Spesifikasi <i>Board</i> Arduino Uno .....	24
<b>Tabel 2.3</b>	Taksiran Orde ke- $n$ dan Nilai Titik Singgung .....	26
<b>Tabel 2.4</b>	Nilai Konstan Perbandingan $T$ .....	27
<b>Tabel 2.5</b>	Pengaruh Parameter $Kp$ , $Ki$ , dan $Kd$ .....	36
<b>Tabel 3.1</b>	Spesifikasi Mesin Pemotong Rumput 2 Tak .....	44
<b>Tabel 3.2</b>	Spesifikasi Motor DC .....	45
<b>Tabel 3.3</b>	Data Hasil Pembacaan terhadap Sensor <i>Rotary Encoder</i> dan <i>Tachometer Digital</i> .....	60
<b>Tabel 3.4</b>	Perbandingan Data Kecepatan Hasil Linearisasi dengan Kecepatan Asli (dalam rpm).....	60
<b>Tabel 3.5</b>	Data Percobaan Konstanta Balik Motor DC .....	63
<b>Tabel 3.6</b>	Data Percobaan Kondisi Maksimal dan Nominal .....	64
<b>Tabel 3.7</b>	Data Hasil Pengukuran Parameter .....	66
<b>Tabel 3.8</b>	Hasil pengukuran parameter $La$ , $Ra$ , $Lf$ , $Rf$ .....	66
<b>Tabel 3.9</b>	Data Hasil Pengukuran Arus Rem Elektromagnetik .....	67
<b>Tabel 3.10</b>	Persamaan Model dan Validasi <i>Plant</i> ICE Beban Nominal .....	81
<b>Tabel 3.11</b>	Pengujian Metode Pendekatan Model .....	82
<b>Tabel 3.12</b>	Nilai Matriks $\mathbf{P}$ Melalui Penyelesaian ARE .....	84
<b>Tabel 3.13</b>	Nilai Parameter $Kp$ , $Ki$ , dan $Kd$ .....	84
<b>Tabel 4.1</b>	Nilai IAE Saat Arus Beban 0,6 A .....	92
<b>Tabel 4.2</b>	Nilai IAE Saat Arus Beban 0,7 A .....	92
<b>Tabel 4.3</b>	Nilai IAE Saat Beban 0.8 A .....	93

*--- Halaman ini sengaja dikosongkan ---*

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	HEV Konfigurasi Paralel .....	6
<b>Gambar 2.2</b>	HEV Konfigurasi Seri .....	7
<b>Gambar 2.3</b>	HEV Konfigurasi Seri-Paralel .....	7
<b>Gambar 2.4</b>	<i>Planetary Gear</i> .....	8
<b>Gambar 2.5</b>	Komponen ICE .....	9
<b>Gambar 2.6</b>	Skema Langkah Kerja Mesin 2 Tak .....	10
<b>Gambar 2.7</b>	Kaidah Tangan Kiri .....	12
<b>Gambar 2.8</b>	Gaya Lorentz pada Motor DC.....	13
<b>Gambar 2.9</b>	Motor DC Sederhana .....	15
<b>Gambar 2.10</b>	Karakteristik Motor DC dan Konfigurasinya .....	16
<b>Gambar 2.11</b>	Bentuk Fisik Rem Elektromagnetik .....	17
<b>Gambar 2.12</b>	Ilustrasi Rem Magnetik Arus <i>Eddy</i> .....	18
<b>Gambar 2.13</b>	Konstruksi <i>Timing Belt</i> .....	19
<b>Gambar 2.14</b>	Konfigurasi <i>Gear</i> Satu Poros .....	19
<b>Gambar 2.15</b>	(a) <i>Pitch Circle</i> , (b) <i>Gear</i> dengan 20 Gigi, <i>Pitch Diameter</i> 1'', dan <i>Diametral Pitch</i> 20 .....	20
<b>Gambar 2.16</b>	Konfigurasi <i>Gear</i> Beda Poros .....	21
<b>Gambar 2.17</b>	Konstruksi <i>Rotary Encoder</i> .....	22
<b>Gambar 2.18</b>	<i>Board</i> Arduino Uno .....	23
<b>Gambar 2.19</b>	Respon <i>Step</i> Sistem dengan waktu $t_1$ , $t_2$ , $t_b$ , $T_U$ , dan $T_N$ .....	25
<b>Gambar 2.20</b>	Konfigurasi Fisik Motor DC Konfigurasi Paralel.....	28
<b>Gambar 2.21</b>	Blok Diagram Fisik Motor DC Konfigurasi Paralel .....	29
<b>Gambar 2.22</b>	Diagram Blok Kontroler PID.....	36
<b>Gambar 2.23</b>	Diagram Blok PID-LQR .....	38
<b>Gambar 3.1</b>	Simulator HEV dengan Konfigurasi Paralel.....	41
<b>Gambar 3.2</b>	Diagram Blok Sistem .....	42
<b>Gambar 3.3</b>	Konfigurasi Perangkat Keras Simulator PHEV .....	43
<b>Gambar 3.4</b>	Bentuk Fisik Mesin Pemotong Rumput 2 Tak pada Simulator HEV .....	45
<b>Gambar 3.5</b>	Bentuk Fisik Motor DC pada Simulator HEV .....	46
<b>Gambar 3.6</b>	Transmisi <i>Gear</i> .....	47
<b>Gambar 3.7</b>	Ilustrasi Sensor Posisi pada Pedal Gas.....	49
<b>Gambar 3.8</b>	Potensiometer pada Simulator HEV .....	49
<b>Gambar 3.9</b>	Konfigurasi Rem Elektromagnetik .....	50
<b>Gambar 3.10</b>	Bentuk Fisik Rem Elektromagnetik.....	50
<b>Gambar 3.11</b>	Bentuk Fisik Sistem Minimum Arduino Uno R3 .....	51

<b>Gambar 3.12</b>	Rangkaian <i>Driver</i> Motor DC.....	52
<b>Gambar 3.13</b>	Bentuk Fisik <i>Driver</i> Motor DC .....	52
<b>Gambar 3.14</b>	Skema Rangkaian Penyearah Gelombang Penuh .....	53
<b>Gambar 3.15</b>	Bentuk Fisik Rangkaian Penyearah Gelombang Penuh pada Simulator HEV.....	53
<b>Gambar 3.16</b>	<i>Rotary Encoder</i> pada Simulator HEV .....	54
<b>Gambar 3.17</b>	Modul <i>Optocoupler</i> dan <i>Encoder Disk</i> .....	54
<b>Gambar 3.18</b>	Skema Rangkaian <i>Optocoupler</i> .....	55
<b>Gambar 3.19</b>	Tampilan Program Arduino.....	56
<b>Gambar 3.20</b>	Tampilan <i>Software</i> LabView.....	57
<b>Gambar 3.21</b>	Tampilan MATLAB R2013a .....	58
<b>Gambar 3.22</b>	Tampilan Simulink pada MATLAB R2013a .....	58
<b>Gambar 3.23</b>	Bentuk Fisik <i>Tachometer Digital</i> .....	59
<b>Gambar 3.24</b>	Respon Motor DC saat Diberikan Input Sinyal <i>Step</i> .....	69
<b>Gambar 3.25</b>	Diagram Blok Motor DC .....	69
<b>Gambar 3.26</b>	Diagram Blok Fisik Motor DC Konfigurasi Paralel .....	70
<b>Gambar 3.27</b>	Diagram Blok Fisik Motor DC Konfigurasi Paralel Hasil Linierisasi .....	71
<b>Gambar 3.28</b>	Diagram Blok Pemodelan Strejc .....	74
<b>Gambar 3.29</b>	Respon Kecepatan Putar ICE Beban Minimal.....	76
<b>Gambar 3.30</b>	Respon Kecepatan Putar ICE Beban Nominal .....	78
<b>Gambar 3.31</b>	Respon Kecepatan Putar ICE Beban Maksimal .....	80
<b>Gambar 3.32</b>	Hasil Identifikasi dan Pemodelan ICE Beban Nominal..	81
<b>Gambar 3.33</b>	Desain Simulink PID-LQR.....	85
<b>Gambar 4.1</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,6 A .....	88
<b>Gambar 4.2</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,6 A.....	88
<b>Gambar 4.3</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,7 A .....	89
<b>Gambar 4.4</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,7 A.....	89
<b>Gambar 4.5</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,8 A .....	90
<b>Gambar 4.6</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,8 A.....	90
<b>Gambar 4.7</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,6 A .....	94
<b>Gambar 4.8</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,6 A....	95
<b>Gambar 4.9</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,6 A .....	95
<b>Gambar 4.10</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,7 A .....	96

<b>Gambar 4.11</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,7 A ...	97
<b>Gambar 4.12</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,7 A.....	97
<b>Gambar 4.13</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,8 A.....	98
<b>Gambar 4.14</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,8 A ...	99
<b>Gambar 4.15</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,8 A.....	99
<b>Gambar 4.16</b>	Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 50 VDC.....	101
<b>Gambar 4.17</b>	Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 57,15 VDC.....	101
<b>Gambar 4.18</b>	Respon Kecepatan Putar HEV Hasil Implementasi Menggunakan Kontroler PID-LQR dengan 3 Variasi Beban .....	102

--- Halaman ini sengaja dikosongkan ---



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jumlah kendaraan bermotor di Indonesia terus mengalami peningkatan. Hal ini menunjukkan betapa besar kontribusi pencemaran udara dari kendaraan transportasi selain dari kegiatan industri dan rumah tangga. Dari sekian banyak penggunaan kendaraan bermotor tersebut maka akan berbanding lurus dengan penggunaan bahan bakar minyak. Penggunaan bahan bakar minyak untuk kendaraan bermotor yang semakin meningkat menyebabkan menipisnya cadangan energi minyak bumi.

Seiring berkembangnya isu efek rumah kaca karena pencemaran udara dan krisis energi, maka pengembang teknologi terus melakukan inovasi. Sektor transportasi juga melakukan inovasi dibidang teknologi kendaraan. *Hybrid Electric Vehicle* (HEV) merupakan suatu kendaraan dengan konsep ramah lingkungan dan hemat energi yang diharapkan menjadi salah satu alternatif menanggulangi efek rumah kaca dan krisis energi.

Kendaraan *Hybrid Electric Vehicle* (HEV) dengan mesin *hybrid* adalah kendaraan yang menggunakan dua atau lebih tenaga sebagai sumber penggerak [1], yang merupakan gabungan antara mesin bakar dan motor listrik. Pada mesin *hybrid*, tenaga yang digunakan untuk menggerakkan kendaraan bisa berasal dari ICE, motor listrik, maupun gabungan diantara keduanya [1]. Kelebihan dari mesin *hybrid* adalah saat ICE tidak mampu mempertahankan kecepatan maka motor listrik akan membantu memberikan tenaga [2]. Ketidakmampuan dalam mempertahankan kecepatan tersebut biasanya muncul saat terjadi pembebanan lebih pada kendaraan.

Metode *PID-Linear Quadratic Regulator* (PID-LQR) digunakan untuk mengatur kecepatan motor listrik, sehingga dapat membantu ICE saat terjadi pembebanan lebih yang dapat menurunkan kecepatan pada kendaraan. Metode PID-LQR merupakan metode PID optimal dengan pendekatan LQR untuk menentukan parameter  $K_p$ ,  $K_i$  dan  $K_d$  pada kontroler PID. Dengan menggunakan metode ini diharapkan terjadinya penurunan kecepatan akibat efek pembebanan berlebih dapat diatasi dengan adanya pembagian kerja antara dua tenaga sebagai sumber penggerak. Pada Tugas Akhir ini digunakan Simulator *Parallel Hybrid*

*Electric Vehicle* (PHEV). Simulator PHEV memadukan dua buah pembangkit torsi untuk menanggung beban yang terjadi, dengan ICE bergerak sebagai penggerak utama dan motor DC bertindak sebagai penggerak pembantu [3]. Simulator PHEV yang digunakan terdiri dari mesin pemotong rumput sebagai ICE, motor DC paralel sebagai motor lisrik, dan rem elektromagnetik sebagai beban pengereman yang telah dirancang sebelumnya pada Tugas Akhir tahun 2010 [4].

## 1.2 Perumusan Masalah

Saat ICE mendapat beban dari rem elektromagnetik maka kecepatan putar ICE akan menurun sehingga kecepatan HEV tidak sesuai dengan *output* yang diharapkan serta mempengaruhi kinerja dari sistem. Efek pembebanan tersebut biasanya terjadi ketika kendaraan berada pada tanjakan. Permasalahan yang menjadi topik dalam Tugas Akhir ini adalah bagaimana mengatur kecepatan motor listrik agar dapat membantu ICE untuk menanggung beban yang berlebih sehingga dapat mencapai kecepatan putar yang sesuai. Oleh karena itu, dibutuhkan suatu kontroler untuk melakukan pengaturan kerja dari motor listrik agar bekerja sesuai dengan kebutuhan yang diinginkan.

## 1.3 Batasan Masalah

Batasan masalah dalam penyelesaian Tugas Akhir ini adalah sebagai berikut :

1. *Plant* yang digunakan adalah simulator HEV yang terdiri dari ICE sebagai penggerak utama yang dioperasikan secara manual tanpa diberikan aksi kontrol, motor listrik yaitu motor DC sebagai penggerak pembantu dan rem elektromagnetik sebagai pemberi efek pembebanan.
2. Sumber tegangan untuk mengoperasikan motor listrik berasal dari jala-jala PLN sehingga tidak ada mekanisme *charging*, *discharging* dan *regenerative braking*.
3. Pengaturan regulasi pada simulator HEV dilakukan oleh motor DC.
4. Motor listrik dioperasikan dengan *range* sumber tegangan 0 sampai 150 V.
5. Model nominal digunakan untuk mengatasi pembebanan pada *range* pembebanan nominal.

## 1.4 Tujuan

Tujuan dari pelaksanaan Tugas Akhir ini adalah merancang kontroler *PID-Linear Quadratic Regulator* (PID-LQR) untuk pengaturan kecepatan pada simulator PHEV sehingga dapat mengatasi permasalahan adanya efek pembebanan berlebih pada HEV. Hasil dari perancangan kontroler diharapkan dapat memperoleh performansi terbaik untuk pengaturan kecepatan pada pembebanan yang berbeda-beda sehingga kecepatan yang diinginkan dapat terus dipertahankan.

## 1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima BAB dengan sistematika penulisan sebagai berikut :

### **BAB 1 PENDAHULUAN**

Bab ini berisi latar belakang, perumusan masalah, tujuan, sistematika penulisan dan relevansi dari Tugas Akhir ini.

### **BAB II DASAR TEORI**

Bab ini berisikan penjelasan mengenai konsep dasar yang berkaitan dengan pengerjaan Tugas Akhir ini.

### **BAB III PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan perangkat keras dan perangkat lunak, pengujian sistem, identifikasi dan pemodelan *plant*, serta perancangan kontroler PID-LQR.

### **BAB IV PENGUJIAN DAN ANALISA**

Bab ini memuat hasil simulasi dan implementasi kontroler PID-LQR dengan kondisi beban yang berubah-ubah dan menganalisa dari data yang telah didapatkan.

### **BAB V PENUTUP**

Berisi pemaparan kesimpulan dan saran dari hasil pembahasan yang telah diperoleh serta sebagai pertimbangan bagi pembaca untuk pengembangan dari Tugas Akhir ini selanjutnya.

## 1.6 Relevansi

Hasil yang diperoleh dari penelitian Tugas Akhir ini diharapkan dapat memberikan manfaat tentang penelitian dalam bidang pengaturan kendaraan *hybrid* baik dalam referensi kontroler pada pengaturan

kecepatan untuk mendapatkan performansi regulasi yang lebih baik. Selain itu, juga membantu pengembangan teknologi otomotif dan industri dalam negeri, memberikan manfaat serta kontribusi bagi dunia pendidikan, dan sebagai ilmu pengetahuan.

## **BAB 2**

### **DASAR TEORI**

#### **2.1 *Hybrid Electric Vehicle (HEV)* [5]**

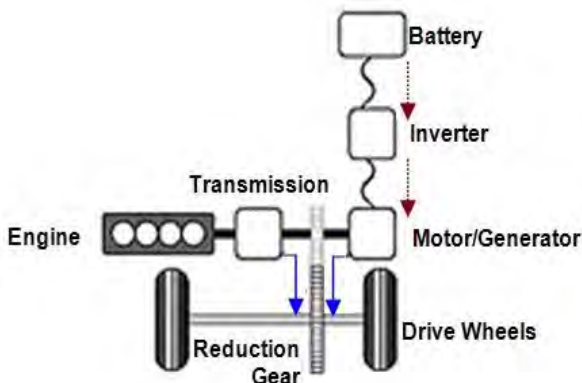
Kendaraan listrik hibrid berkembang dengan pesat di era 1990-an. Diinisiasi oleh Toyota dengan produk kenamaannya, Toyota Prius, dan kini telah banyak perusahaan mobil yang memproduksi kendaraan listrik hibrid. Kendaraan jenis ini menjadi sangat populer karena kemampuannya menghemat penggunaan bahan bakar yang saat ini masih didominasi oleh bahan bakar fosil. Selain itu, bumi dengan perubahan suhu yang semakin ekstrim mendesak manusia untuk berusaha lebih keras lagi mengurangi emisi CO<sup>2</sup>.

Pada dasarnya, kendaraan listrik hibrid bekerja dengan mengkombinasikan pemanfaatan mesin bakar dan motor listrik untuk mencapai penggunaan bahan bakar yang lebih ekonomis. Pemanfaatan motor listrik diharapkan dapat membantu mesin bakar bekerja dengan temperatur mesin yang tidak terlalu panas. Hal ini dicapai dengan menjaga mesin bakar bekerja dengan kecepatan yang relatif konstan. Selain itu, sistem pengereman regeneratif pada kendaraan listrik hibrid, dapat memberikan suplai listrik ke baterai sehingga energi pengereman tidak terbuang sia-sia sebagai panas seperti pada kendaraan konvensional. Bantuan dari motor listrik diberikan saat kendaraan menanjak, menurun, percepatan, mulai berjalan dan perlambatan. Saat kendaraan mulai berjalan, mesin bakar dinyalakan. Pada kecepatan mencapai 1000 rpm, motor listrik dimatikan dan mesin bakar digunakan. Saat kendaraan berada di turunan, mesin bakar mengisi baterai jika baterai dalam tidak terisi penuh. Saat percepatan atau di tanjakan, motor listrik dan mesin bakar menyuplai tenaga gerak secara bersamaan (pada konfigurasi seri: sumber energi motor listrik didapat dari baterai dan mesin bakar). Saat perlambatan, pengereman regeneratif mengisi baterai.

##### **2.1.1 HEV Konfigurasi Paralel [3] [6]**

Pada konfigurasi paralel, terdapat dua jalur aliran daya, aliran mekanik dan aliran elektrik. Setiap aliran daya bisa menjalankan kendaraan secara individu maupun bersama. Motor listrik bisa difungsikan sebagai motor maupun generator. Pengaturan kerja motor listrik sangat penting sehingga tidak memboroskan energi yang tersimpan di dalam baterai. Kelemahan konfigurasi ini adalah kapasitas

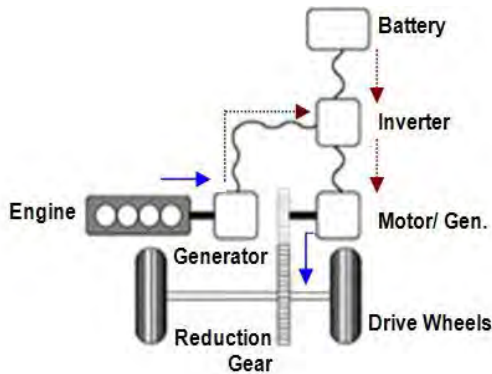
*charge* baterai tidak terlalu besar dan bersifat diskontinyu serta kesulitannya menentukan kondisi kerja mesin bakar yang optimal karena langsung terhubung ke sistem transmisi. Namun di sisi lain, tenaga gerak yang dirasakan dapat setara dengan kendaraan konvensional. Konfigurasi paralel lebih cocok diterapkan pada kendaraan yang jarak tempuhnya jauh (fluktuasi kecepatannya relatif rendah) atau yang membutuhkan daya yang besar (efisiensi meningkat seiring dengan peningkatan daya). HEV konfigurasi paralel dapat dilihat pada **Gambar 2.1**.



**Gambar 2.1** HEV Konfigurasi Paralel [6]

### 2.1.2 HEV Konfigurasi Seri [3] [6]

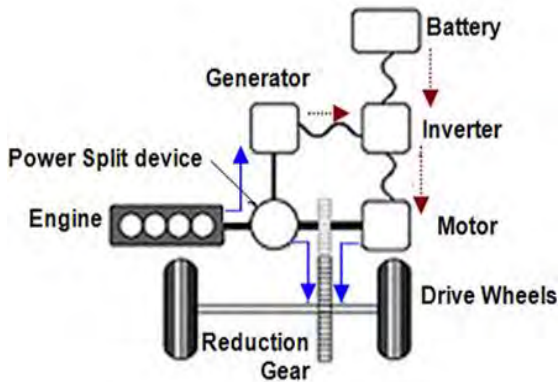
Konfigurasi seri seperti yang ditunjukkan pada **Gambar 2.2**, hanya menggunakan motor listrik sebagai penggerak yang terhubung langsung dengan sistem transmisi. Tenaga gerak dari mesin bakar selalu diubah ke dalam listrik oleh generator listrik sehingga efisiensi sistem sangat bergantung pada efisiensi generator dan motor listrik yang digunakan. Daya listrik yang dihasilkan generator dapat menyuplai motor listrik, jika dibutuhkan. Jika tidak, akan digunakan untuk mengisi baterai. Karena operasi mesin bakar tidak tergantung dari kecepatan kendaraan dan beban, maka mesin bakar bisa beroperasi di kondisi optimal sepanjang waktu. Kekurangan dari konfigurasi ini adalah respon yang tidak terlalu cepat bila diinginkan akselerasi tiba-tiba. Konfigurasi seri sangat cocok untuk penggunaan kendaraan dengan frekuensi jalan dan berhenti yang tinggi seperti pada bis-bis di perkotaan.



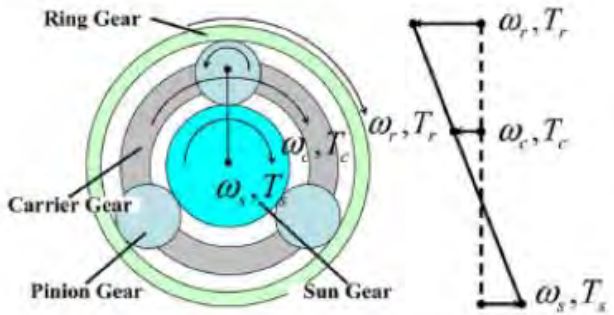
**Gambar 2.2** HEV Konfigurasi Seri [6]

### 2.1.3 HEV Konfigurasi Seri-Paralel [3] [6]

HEV dengan konfigurasi seri – paralel menggunakan *power split device* seperti yang ditunjukkan pada **Gambar 2.3**, sehingga dapat disebut konfigurasi *power split*. Dengan adanya *power split device* kekurangan pada konfigurasi paralel dan seri dapat teratasi. Sistem ini dikembangkan oleh Toyota pertama kali pada Toyota Prius generasi pertama. Prinsip dasar dari *power split device* ini adalah menggunakan sistem *planetary gear* seperti pada **Gambar 2.4**.



**Gambar 2.3** HEV Konfigurasi Seri-Paralel [6]



**Gambar 2.4** Planetary Gear [3]

Planetary gear terdiri dari tiga komponen, antara lain: *sun gear* yang berada di pusat, *ring gear* yang berada di posisi paling luar, *pinion gear* yang menghubungkan *sun gear* dan *ring gear*, dan *carrier gear* yang menghubungkan semua *pinion gear*.

Dari **Gambar 2.4** dapat diamati bahwa *sun gear* terhubung dengan generator, *planetary carrier* terhubung dengan motor bakar, dan *ring gear* terhubung dengan motor listrik atau *shaft* utama. Desain yang demikian, memungkinkan pengisian baterai secara kontinu dari ICE sekalipun sedang digunakan untuk menggerakkan kendaraan. Jadi apabila tiba-tiba dibutuhkan akselerasi, maka respon kendaraan akan cepat dan kebutuhan daya motor listrik terus terjaga.

Berikut adalah perbandingan kapasitas konfigurasi paralel, seri, dan *power split device* (seri-paralel) dapat dilihat pada **Tabel 2.1**.

**Tabel 2.1** Perbandingan Kapasitas Konfigurasi HEV [3]

Type	Efisiensi bahan bakar			Performansi	
	Idling Stop	Recovery Energy	Efisiensi	Akselerasi	Kestabilan kecepatan
Seri	●	◎	●	○	○
Paralel	●	●	●	●	○
Seri-Paralel	◎	◎	◎	●	●



Sangat baik



Baik



Tidak Baik

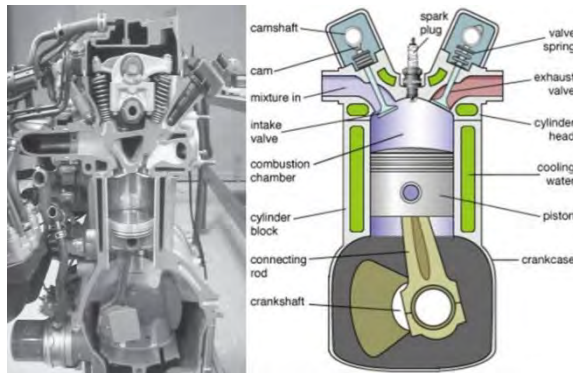


## 2.2 Simulator Hybrid Electric Vehicle

Pada penelitian ini digunakan simulator *Hybrid Electric Vehicle* dengan konfigurasi paralel yang terdiri dari ICE 2 tak sebagai penggerak utama, motor DC sebagai penggerak pendukung, dan beban berupa rem elektromagnetik.

### 2.2.1 Internal Combustion Engine (ICE) 2 Tak [7]

ICE merupakan mesin pembakaran dalam dimana langkah kerjanya beroperasi berdasar prinsip satu siklus yang bekerja secara terus menerus. ICE dibagi berdasarkan jumlah *stroke* atau kayuhan (tak). Jumlah *stroke* menentukan jumlah proses yang dilakukan dalam satu siklus. Komponen ICE dapat dilihat pada **Gambar 2.5**.



**Gambar 2.5** Komponen ICE [7]

Menurut langkah kerjanya ICE dibagi menjadi mesin dengan proses 2 langkah (tak) dan mesin dengan proses 4 langkah (tak). Mesin 2 tak adalah mesin pembakaran dalam yang dalam satu siklus pembakaran terjadi dua langkah piston, berbeda dengan mesin 4 tak yang mempunyai empat langkah piston dalam satu siklus pembakaran, meskipun keempat proses (*intake*, kompresi, tenaga, pembuangan) juga terjadi. ICE 2 tak menyelesaikan keempat tahap siklus, yaitu *intake stroke*, *compression stroke*, *power stroke*, dan *exhaust stroke* hanya dalam dua langkah (kayuhan).

Untuk memahami prinsip kerjanya, perlu dimengerti istilah baku yang berlaku dalam teknik otomotif :

- a. TMA (Titik Mati Atas) atau TDC (*Top Dead Centre*), posisi piston berada pada titik paling atas dalam silinder mesin atau piston berada pada titik paling jauh dari poros engkol (*crankshaft*).
- b. TMB (Titik Mati Bawah) atau BDC (*Bottom Dead Centre*), posisi piston berada pada titik paling bawah dalam silinder mesin atau piston berada pada titik paling dekat dengan poros engkol (*crankshaft*).
- c. Ruang bilas yaitu ruangan dibawah piston dimana terdapat poros engkol (*crankshaft*), sering disebut dengan bak engkol (*crankcase*) berfungsi gas hasil campuran udara, bahan bakar dan pelumas bisa tercampur lebih merata.
- d. Pembilasan (*scavenging*) yaitu proses pengeluaran gas hasil pembakaran dan proses pemasukan gas untuk pembakaran dalam ruang bakar.



**Gambar 2.6** Skema Langkah Kerja Mesin 2 Tak [7]

Prinsip kerja dari ICE dengan siklus 2 langkah atau mesin 2 tak seperti yang ditunjukkan pada **Gambar 2.6** dapat dijelaskan sebagai berikut :

- a. **Langkah ke-1 (Isap dan Kompresi) :**
  - a. Piston bergerak dari TMA ke TMB.
  - b. Saat bergerak dari TMA ke TMB, piston akan menekan ruang bilas yang berada di bawahnya. Semakin jauh piston

meninggalkan TMA menuju TMB akan semakin meningkat pula tekanan di ruang bilas.

- c. Pada titik tertentu, piston (*ring piston*) akan melewati lubang pembuangan gas dan lubang pemasukan gas. Posisi masing-masing lubang tergantung dari desain perancang. Umumnya *ring piston* akan melewati lubang pembuangan terlebih dahulu.
- d. Pada saat *ring piston* melewati lubang pembuangan, gas di dalam ruang bakar keluar melalui lubang pembuangan.
- e. Pada saat *ring piston* melewati lubang pemasukan, gas yang tertekan di dalam ruang bilas akan terpompa masuk ke dalam ruang bakar, sekaligus mendorong keluar gas yang ada di dalam ruang bakar menuju lubang pembuangan.
- f. Piston terus menekan ruang bilas sampai titik TMB, sekaligus memompa gas dalam ruang bilas menuju ke dalam ruang bakar.

**b. Langkah ke-2 (Pembakaran dan Buang) :**

- a. Piston bergerak dari TMB ke TMA.
- b. Saat bergerak dari TMB ke TMA, piston akan menghisap gas hasil percampuran udara, bahan bakar dan pelumas ke dalam ruang bilas. Percampuran ini dilakukan oleh karburator atau sistem injeksi.
- c. Saat melewati lubang pemasukan dan lubang pembuangan, piston akan mengkompresi gas yang terjebak di dalam ruang bakar.
- d. Piston akan terus mengkompresi gas dalam ruang bakar sampai TMA.
- e. Beberapa saat sebelum piston sampai di TMA, busi akan menyala untuk membakar gas dalam ruang bakar. Waktu nyala busi tidak terjadi saat piston sampai ke TMA, melainkan terjadi sebelumnya. Ini dimaksudkan agar puncak tekanan akibat pembakaran dalam ruang bakar bisa terjadi saat piston mulai bergerak dari TMA ke TMB, karena proses pembakaran membutuhkan waktu untuk bisa membuat gas terbakar dengan sempurna oleh nyala api busi.

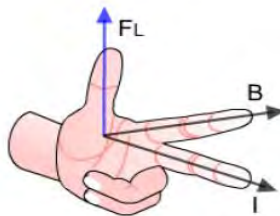
Karena mesin 2 tak dalam 1 putaran kruk as/*crankshaft* melaksanakan 4 siklus, mesin 2 tak ini lebih responsif dan akselerasinya bagus. Akan tetapi, mesin ini mengeluarkan tenaga yang besar pada saat putaran/rpm tinggi sehingga membuat mesin ini membutuhkan bahan bakar yang relatif banyak. Akan tetapi, mesin ini menghasilkan tenaga yang lebih besar dibandingkan mesin 4 tak. Bahan bakar mesin ini tak

hanya bensin, tetapi dapat dioplos dengan oli khusus yang biasa disebut oli samping untuk melumasi bagian dalam mesin sehingga oli mesin hanya melumasi bagian transmisi. Hal ini menyebabkan mesin 2 tak mengeluarkan asap karena membakar oli samping.

### 2.2.2 Motor Listrik DC [2] [8]

Mesin listrik adalah alat yang dapat mengubah energi mekanik menjadi energi listrik atau dari energi listrik menjadi energi mekanik. Ketika mesin listrik mengubah energi mekanik menjadi energi listrik menggunakan alat konversi yang disebut generator. Sebaliknya dengan alat konversi yang disebut motor, energi listrik dapat diubah menjadi energi mekanik. Motor listrik bekerja dengan memanfaatkan gaya-gaya magnet. Motor listrik dibagi menjadi dua jenis, yaitu motor arus bolak-balik (AC) dan motor arus searah (DC).

Prinsip kerja motor DC berawal dari prinsip bahwa apabila terdapat konduktor yang menghantarkan arus dan diletakkan pada medan magnet maka akan menghasilkan suatu gaya. Arah gaya yang dihasilkan ditentukan melalui kaidah tangan kiri seperti dapat dilihat pada **Gambar 2.7**, dengan ibu jari diibaratkan sebagai arah gaya, jari telunjuk sebagai arah medan magnet, dan jari tengah diibaratkan sebagai arah arus yang mengalir pada konduktor.



**Gambar 2.7** Kaidah Tangan Kiri [8]

Medan magnet menghasilkan magnet permanen yang nilainya konstan, sedangkan komutator dan sikat berfungsi menyalurkan alur listrik dari sumber di luar motor ke dalam kumparan jangkar.

Medan *stator* menghasilkan  $\phi$  dari kutub *U* ke kutub *S*. Sikat arang menyentuh terminal kumparan *rotor* di bawah kutub. Bila sikat arang dihubungkan pada satu sumber arus searah di luar tegangan *V*, maka satu arus *I* masuk ke terminal kumparan *rotor* di bawah kutub *U* dan keluar dari terminal di bawah kutub *S*. Dengan adanya fluks *stator*

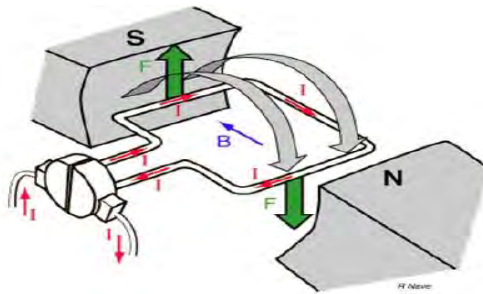
dan arus *rotor* akan menghasilkan satu gaya  $F$  bekerja pada kumparan yang dikenal dengan gaya Lorentz. Arah  $F$  yang ditunjukkan pada **Gambar 2.8** menghasilkan torsi yang memutar *rotor* ke arah yang berlawanan dengan jarum jam. Besar gaya Lorentz yang dihasilkan dapat ditentukan berdasarkan **Persamaan (2.1)**.

$$F = BIL\sin\theta \quad (2.1)$$

dengan,

- $F$  : Gaya Lorentz (N)
- $B$  : Kerapatan fluks magnet (Weber/ $m^2$ )
- $I$  : Arus yang mengalir (A)
- $L$  : Panjang kawat konduktor (m)
- $\theta$  : Sudut yang terbentuk antara arah arus medan dengan arah arus yang mengalir pada kawat (Derajat)

Saat arus mengalir melewati kawat yang memotong medan magnet seperti pada **Gambar 2.8**, maka akan timbul gaya Lorentz. Gaya Lorentz inilah yang menimbulkan torsi pada *shaft* motor. Karena motor DC membangkitkan torsi, maka seringkali disebut sebagai pembangkit torsi.



**Gambar 2.8** Gaya Lorentz pada Motor DC [2]

Komutator berfungsi untuk membalik polaritas, sehingga arus mengalir dengan arah yang berlawanan dari sebelumnya. Perhatikan **Gambar 2.8**, saat kawat berputar sesuai arah gaya hingga posisi tegak lurus gaya yang bekerja tetap sama. Namun setelah itu diperlukan gaya yang berlawanan agar kawat tetap berputar. Untuk membalik gaya,

maka arah arus perlu dibalik polaritasnya dari kedua ujung kawat. Komutator akan terbalik dengan sendirinya setelah kawat berputar selama setengah putaran. Komutator secara fisik berupa segmen-segmen, biasanya terbuat dari tembaga dan terletak pada rotor. Sikat arang (*brushes*) terbuat dari karbon, ditekan pada komutator dengan menggunakan pegas. Sikat arang ini membawa arus listrik dari sumber menuju kumparan jangkar. Pada komutator, setiap segmen dipisahkan oleh isolator seperti mika atau *polymer*. Setiap segmen juga dihubungkan dengan beberapa kumparan pada kumparan jangkar.

Struktur fisik dari motor DC terdiri dari dua bagian yaitu *stator* (bagian yang diam) dan *rotor* (bagian yang berputar). Pada bagian *stator* merupakan tempat kumparan medan yang berfungsi untuk menghasilkan fluks magnet sedangkan pada bagian *rotor* terdapat rangkaian jangkar seperti kumparan jangkar (*armature*) yang menghasilkan gaya putar, komutator dan sikat. Kumparan jangkar pada dasarnya adalah kawat yang dialiri arus, sehingga menimbulkan gaya Lorentz. Disebut kumparan, karena jumlah kawatnya yang banyak. Dengan demikian semakin banyak kawat, maka gaya Lorentz yang dihasilkan akan semakin besar dan kontinyu. Torsi yang terbentuk mengikuti **Persamaan (2.2)** berikut:

$$\tau = KI_a\phi \quad (2.2)$$

dengan,

- $\tau$  : Torsi yang dibangkitkan (Nm)
- $K$  : Konstanta mesin
- $I_a$  : Arus jangkar (Ampere)
- $\phi$  : Fluks magnet (Weber)

Bila kumparan jangkar dari motor berputar dalam medan magnet dan memotong fluks utama, sesuai dengan hukum induksi elektromagnetik, maka pada kumparan jangkar akan timbul gaya gerak listrik (ggl) induksi yang arahnya sesuai dengan kaidah tangan kanan, di mana arahnya berlawanan dengan tegangan yang diberikan kepada jangkar atau tegangan terminal. Karena arahnya melawan, maka ggl induksi ini disebut ggl lawan dengan **Persamaan (2.3)**.

$$E_a = cn\phi \quad (2.3)$$

dengan,

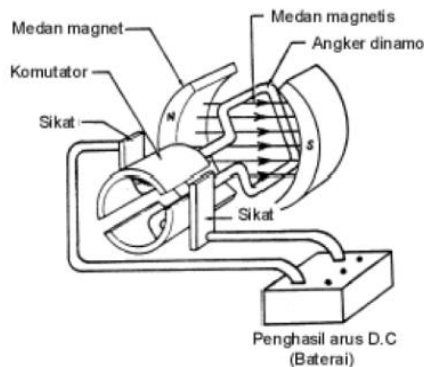
$E_a$  : Gaya gerak listrik induksi (Volt)

$n$  : Kecepatan putaran (rpm)

$\phi$  : Fluksi setiap kutub (Weber)

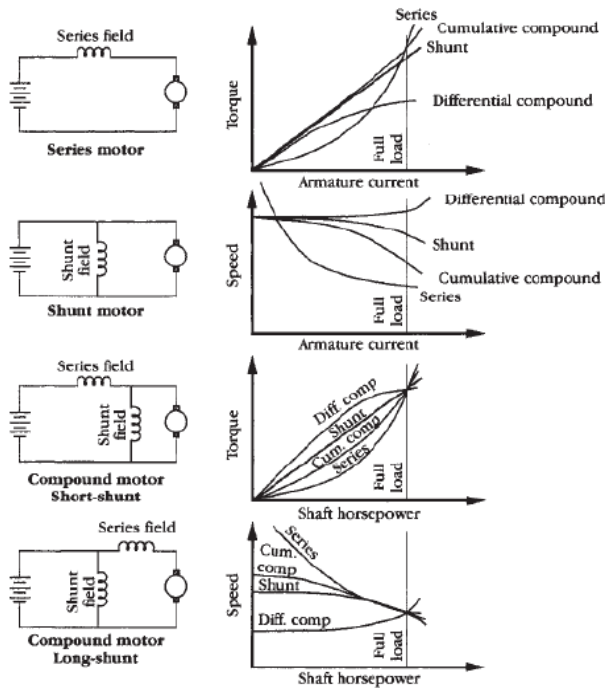
$c$  : Konstanta

Berdasarkan konstruksi nya, motor DC dapat dibedakan menjadi *brushed*, *brushless* dan *permanent magnet*. Bentuk motor DC yang paling sederhana adalah memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen yang dapat dilihat pada **Gambar 2.9**. Yang sering digunakan adalah motor DC *brushed* karena konstruksinya yang sederhana, sehingga berharga lebih murah dan juga memiliki kemampuan yang cukup baik. Motor DC *brushed* terdiri dari beberapa konfigurasi, yaitu paralel, seri, dan *compound* (campuran) yang masing masing memiliki karakteristik dan konfigurasi yang berbeda-beda (dapat dilihat pada **Gambar 2.10**).



**Gambar 2.9** Motor DC Sederhana [8]

Untuk mengatur kecepatan ataupun torsi pada motor DC, terlebih dahulu harus mengetahui dan mengerti mengenai karakteristik tipe motor DC yang akan digunakan. Pada setiap konfigurasi motor DC, memiliki karakteristik dan penurunan persamaan model matematika yang berbeda pula. Dalam pemilihan penggunaan motor DC, maka harus diperhatikan pula konfigurasi dan karakteristik dari setiap konfigurasi.



**Gambar 2.10** Karakteristik Motor DC dan Konfigurasiya [2]

### 2.2.3 Rem Elektromagnetik [2] [8]

Sistem pengereman adalah sebuah sistem yang berfungsi untuk menghalangi suatu gerakan. Sistem pengereman bertugas mengkonversi energi mekanis (energi gerak) suatu benda ke bentuk lain sehingga gerakan benda tersebut menjadi berkurang. Pada mobil yang sedang bergerak misalnya, sistem rem mengkonversikan energi gerak mobil menjadi panas yang terbuang melalui gesekan pada kanvas rem dengan piringan ataupun tromol roda. Dengan kata lain sistem rem konvensional (sebut saja rem tromol ataupun rem cakram) membuang begitu saja energi panas yang terjadi pada saat proses pengereman.

Terdapat dua tipe rem, yaitu tipe rem mekanik dan elektromagnetik. Tipe rem mekanik menggunakan gaya gesek yang menimbulkan gaya lawan terhadap gaya gerak, tetapi tipe rem ini memiliki kekurangan antara lain bisa menimbulkan panas yang berlebihan sehingga



mengurangi gaya gesek yang terjadi. Tipe rem elektromagnetik memanfaatkan gesekan dua permukaan untuk menghasilkan gaya lawan terhadap gaya gerak, yaitu menggunakan gaya magnet untuk menimbulkan gaya lawan. Rem elektromagnetik disebut juga rem magnetik arus *eddy*. Saat ini penggunaannya sudah meluas karena lebih menghemat biaya pemeliharaan dan penggantian komponen yang bergesekan. Bentuk fisik rem elektromagnetik dapat dilihat pada **Gambar 2.11**.

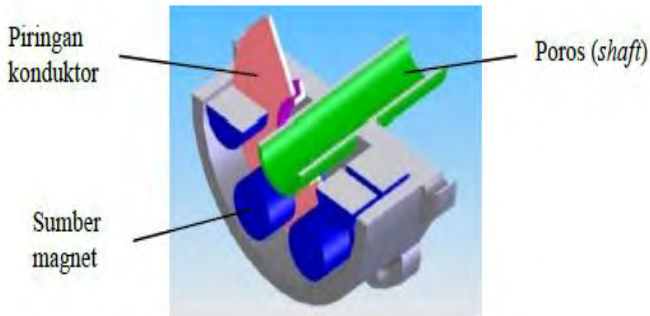
Prinsip dasar dari rem magnetik arus *eddy* menggunakan hukum Faraday dan hukum Lenz. Pada hukum Faraday dijelaskan bahwa besar ggl induksi atau arus induksi tergantung pada laju perubahan fluks dan banyaknya lilitan. Pada hukum Lenz dijelaskan bahwa arus induksi yang timbul arahnya sedemikian sehingga menimbulkan medan magnet induksi yang melawan arah perubahan medan magnet. Dari penjelasan ini dapat disimpulkan bahwa arus induksi akan timbul jika terdapat perubahan fluks. Perubahan fluks ini akan terjadi apabila konduktor tersebut berputar. Semakin cepat putaran maka akan semakin besar pula laju perubahan fluks sehingga arus induksi yang dihasilkan akan semakin besar, begitu juga sebaliknya. Arus induksi yang dihasilkan tadi akan menimbulkan medan magnet induksi yang berlawanan arah dengan arah perubahan medan magnet sehingga akan menjadikan putaran dari konduktor melambat.



**Gambar 2.11** Bentuk Fisik Rem Elektromagnetik [2]

Komponen dasar rem magnetik arus *eddy* adalah sumber magnet yang diposisikan tetap dan konduktor diamagnetik (tidak memiliki sifat magnet namun memiliki sifat penghantar listrik) yang terhubung dengan

poros. Arus *eddy* yang timbul pada sebuah cakram yang dikelilingi medan magnet dapat dilihat pada **Gambar 2.12**.

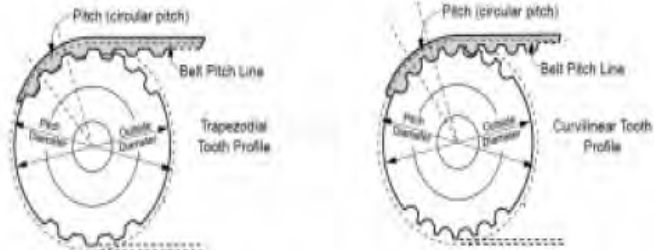


**Gambar 2.12** Ilustrasi Rem Magnetik Arus *Eddy* [8]

#### 2.2.4 *Timing Belt* [2] [9]

*Belt* mempunyai fungsi yang dapat dikatakan vital, karena bila putus maka AC, kelistrikan dan mesin bisa *drop* bahkan mati. *Belt* tersebut diharuskan untuk kuat, tahan lama, keras sekaligus elastis. *Belt* sering kali digunakan untuk menghubungkan dua buah roda gigi motor untuk merubah kecepatan ataupun kecepatan sudut yang bertujuan untuk memperbesar maupun memperkecil nilainya dengan mengatur ukuran roda gigi yang digunakan.

Salah satu jenis *belt* yang sering digunakan untuk permasalahan aplikasi perpindahan gerak adalah *timing belt*. Konstruksi *timing belt* dapat dilihat pada **Gambar 2.13**. *Timing belt* merupakan *belt* yang bertugas menyelaraskan dan menyalurkan putaran kruk as dengan noken as, sehingga proses buka dan tutup klep sesuai dengan posisi piston dan *timing* pengapian busi. *Timing belt* penting pada sebuah mesin, bila putus atau rusak mesin dapat seketika mati. Oleh karena fungsinya yang vital maka *timing belt* ini dibuat dari material yang lebih kuat. *Timing belt* memiliki gerigi – gerigi yang seragam, sehingga lebih baik dari pada menggunakan V-*belt* yang rawan terhadap selip. Pemilihan *timing belt*, terdapat beberapa spesifikasi yang perlu diperhatikan diantaranya gerigi yang lebih panjang akan mencegah adanya selip, konstruksi yang lebih ringan akan mengurangi rugi – rugi sentrifugal, daerah kontak (daerah pertemuan gerigi *belt* dan *gear*) yang lebih besar akan mengurangi tekanan unit, selain itu bagian persimpangan gerigi yang lebih besar akan menyebabkan kekuatan geser yang lebih besar.



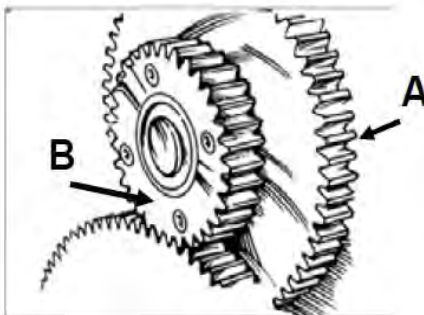
**Gambar 2.13** Konstruksi *Timing Belt* [9]

#### 2.2.4.1 Perbandingan Gear

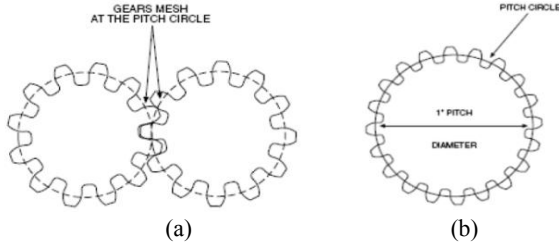
*Gear* merupakan suatu lingkaran yang memiliki gerigi di bagian luarnya. *Gear* ini digunakan untuk mengubah kecepatan sudut atau torsi. Terdapat dua konfigurasi dasar penggunaan *gear* yaitu pada satu poros dan beda poros.

##### a. Konfigurasi *Gear* Satu Poros (*Compound Gear*)

Jumlah pada *gear* dan *diameter pitch* merupakan dua hal yang harus diperhatikan dalam pemilihan kombinasi *gear* yang akan digunakan. Konfigurasi *gear* dengan sistem satu poros terlihat pada **Gambar 2.14**. Ketika dua buah *gear* bersinggungan terdapat lingkaran khayal yang disebut dengan *pitch circle* seperti pada **Gambar 2.15**. Diameter pada *pitch circle* adalah *pitch diameter*, dan rasio jumlah gigi terhadap *pitch diameter* adalah *pitch circle*.



**Gambar 2.14** Konfigurasi *Gear* Satu Poros [2]



**Gambar 2.15** (a) *Pitch Circle*, (b) *Gear* dengan 20 Gigi, *Pitch Diameter* 1'', dan *Diametral Pitch* 20 [2]

b. Konfigurasi *Gear* Beda Poros

*Gear* beda poros disebut juga dengan *gear* reduksi. Pada konfigurasi *gear* beda poros, *gear* menghubungkan dua poros yang berbeda dengan menggunakan *belt*, *chain*, ataupun rantai. Konfigurasi *gear* beda poros dapat dilihat pada **Gambar 2.16**. Konfigurasi dasar untuk dua *gear* yang berbeda poros dapat dilihat pada **Persamaan (2.4)**.

$$V_A = V_B \quad (2.4)$$

$$\omega_A R_A = \omega_B R_B$$

$$\frac{\omega_A}{\omega_B} = \frac{R_B}{R_A} \quad (2.5)$$

Misalkan  $R_A = 4R_B$ , maka

$$\omega_A = \frac{1}{4} \omega_B \quad (2.6)$$

Torsi pada *gear* dapat dihitung melalui **Persamaan (2.7)**.

$$\tau = F \cdot R \quad (2.7)$$

dengan  $\tau$  merupakan torsi (Nm),  $F$  merupakan gaya yang bekerja pada *gear* dan  $R$  merupakan jari-jari *gear*. Dari **Persamaan (2.6)** maka didapatkan aturan gaya yang bekerja pada *gear* adalah sebagai berikut :

$$F_A = F_B \quad (2.8)$$

$$\frac{\tau_A}{R_A} = \frac{\tau_B}{R_B}$$

$$\frac{\tau_A}{4R_B} = \frac{\tau_B}{R_B}$$

sehingga didapatkan perbandingan torsi,

$$\frac{\tau_A}{\tau_B} = \frac{4R_B}{R_B}$$

$$\tau_A = 4 \tau_B \quad (2.9)$$

dengan,

$V_A$  : Kecepatan linier pada *gear* A

$V_B$  : Kecepatan linier pada *gear* B

$\omega_A$  : Kecepatan sudut pada *gear* A

$\omega_B$  : Kecepatan sudut pada *gear* B

$R_A$  : Jari-jari pada *gear* A

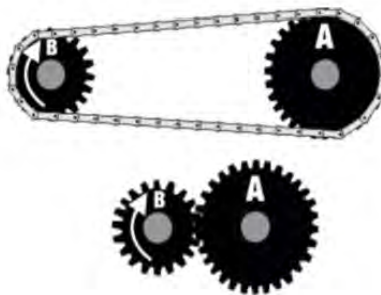
$R_B$  : Jari-jari pada *gear* B

$F_A$  : Gaya yang bekerja pada *gear* A

$F_B$  : Gaya yang bekerja pada *gear* B

$\tau_A$  : Torsi pada *gear* A

$\tau_B$  : Torsi pada *gear* B

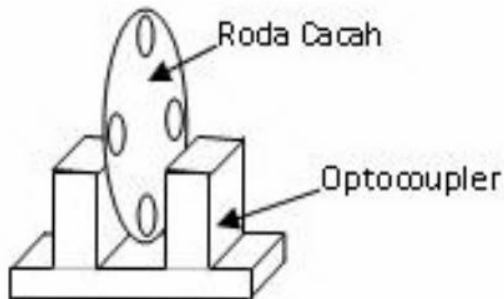


**Gambar 2.16** Konfigurasi *Gear* Beda Poros [2]

### 2.2.5 Rotary Encoder [10]

*Rotary encoder* dikenal sebagai salah satu sensor kecepatan. *Rotary encoder* adalah *device* elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* dapat dibuat dengan menggunakan sensor optik berupa *optocoupler* tipe “U” dan sebuah roda cacah berupa suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. *Optocoupler* merupakan komponen *optoisolator* yang memiliki karakteristik dimana penerima (*photo-transistor*) akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED inframerah) untuk penerima.

Prinsip kerja dari *rotary encoder* adalah ketika ada benda yang berada di antara celah sensornya, maka cahaya yang dikirimkan tidak bisa diterima oleh bagian penerimanya, sehingga menghasilkan tegangan keluaran yang nilainya mendekati VCC, begitu juga sebaliknya, jika tidak ada benda diantara celah sensornya maka akan menghasilkan tegangan keluaran yang nilainya mendekati 0 Volt. Sistem kerja dari *rotary encoder* adalah dengan mendeteksi adanya pulsa yang dihasilkan oleh terdeteksinya lubang pada lempengan yang dipasang pada poros. Pulsa yang dideteksi oleh sensor adalah berupa pulsa *high*. Banyaknya pulsa yang terdeteksi oleh sensor akan langsung dikonversi menjadi rpm dengan bantuan mikrokontroler. Konstruksi *rotary encoder* dapat dilihat pada **Gambar 2.17**.

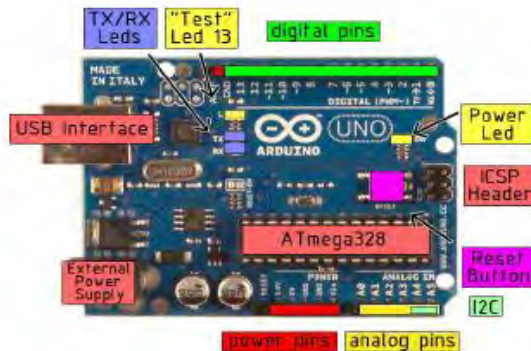


**Gambar 2.17** Konstruksi *Rotary Encoder* [10]

## 2.3 Sistem Minimum Arduino Uno R3 [11]

Arduino Uno adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino adalah sebuah *board* mikrokontroler yang berbasis ATmega328.

Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 *analog input*, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Arduino mampu *support* mikrokontroler; dan dapat dikoneksikan dengan komputer menggunakan kabel USB. *Board* Arduino Uno dapat dilihat pada **Gambar 2.18**.



**Gambar 2.18** *Board* Arduino Uno [11]

Arduino Uno berbeda dari semua *board* Arduino sebelumnya, Arduino Uno tidak menggunakan *chip driver* FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke *serial*. Revisi ke-2 dari *board* Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke *ground*, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode.

Arduino Uno memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, arduino yang lain, ataupun dengan mikrokontroler yang lain. ATmega328 yang digunakan pada Arduino Uno menyediakan komunikasi *serial* UART TTL (5V) yang tersedia pada pin *digital* 0 (RX) dan 1 (TX). Koneksi USB untuk menghubungkan dengan perangkat lain menggunakan ATmega16U2 yang mengacu sebagai *virtual com port* pada komputer. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. *Software* Arduino juga menyediakan *serial monitor* yang memungkinkan data teks dapat terkirim dan diterima oleh arduino dengan mudah. LED RX dan TX pada *board* Arduino akan

menyala saat data sedang dikirim atau diterima melalui kabel USB ke komputer (tetapi tidak menyala untuk komunikasi pada pin *digital* 0 dan 1). Spesifikasi *board* Arduino Uno dapat dilihat pada **Tabel 2.2**.

**Tabel 2.2** Spesifikasi *Board* Arduino Uno [10]

Spesifikasi	Keterangan
Mikrokontroler	ATmega328
Tegangan Operasional	5 V
Tegangan <i>Input</i> (rekomendasi)	7-12 V
Tegangan <i>Input</i> (Limit)	6-20 V
Pin I/O digital	14 (6 diantaranya bisa digunakan untuk <i>output</i> PWM)
Pin Analog <i>Input</i>	6
Arus DC per Pin I/O	40 mA
Arus DC untuk pin 3.3 V	50 mA
<i>Flash Memory</i>	32 KB dengan 0,5 KB digunakan sebagai <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Kecepatan <i>Clock</i>	16 MHz

Arduino Uno dapat diprogram dengan *software* Arduino. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Lingkungan *open-source Arduino* memudahkan untuk menulis kode dan meng-*upload* ke *board* Arduino. *Software* Arduino dapat berjalan pada Windows, Mac OS X, dan Linux, dan perangkat lunak sumber terbuka lainnya.

## 2.4 Identifikasi Sistem [8]

Identifikasi sistem merupakan suatu proses untuk mendapatkan parameter-parameter dari suatu sistem berdasarkan hasil pengukuran *input output plant*. Berdasarkan hasil pengukuran *input output plant* tersebut, maka model matematis suatu *plant* dapat diketahui. Identifikasi dapat dilakukan pada sistem *closed loop* maupun *open loop*. Identifikasi *closed loop* lebih sulit dilakukan, namun dapat mengatasi sistem yang tidak pasti, tidak terpengaruh non linearitas, dan ketelitian tetap terjaga. Sedangkan identifikasi *open loop* lebih sederhana dan mudah dilakukan,



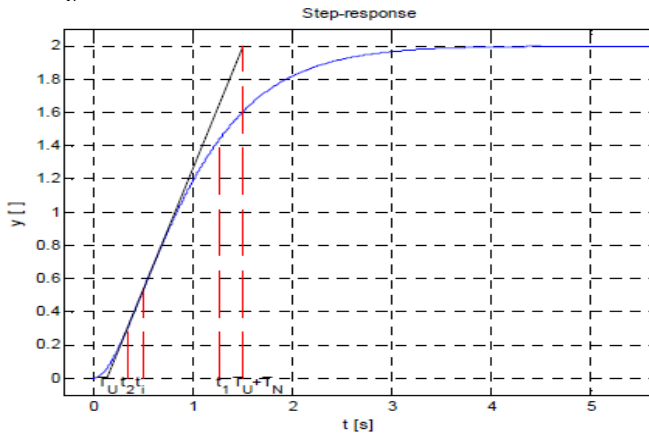
namun sulit diterapkan untuk sistem yang memiliki ketidakpastian respon. Identifikasi *open loop* merupakan salah satu identifikasi yang paling banyak digunakan. Dengan masukan *step*, *impuls*, atau *ramp*, maka respon transien dapat dicari melalui grafik respon sistem.

Pada Tugas Akhir ini identifikasi sistem dilakukan dengan menggunakan identifikasi Strejc untuk *plant* ICE dan identifikasi fisik dengan pengukuran parameter untuk *plant* motor DC.

#### 2.4.1 Identifikasi Strejc [12]

Strejc merupakan salah satu metode identifikasi dengan penarikan garis singgung pada respon *plant*. Pada metode Strejc penarikan garis berdasar pada titik belok (*inflection point*) dari grafik respon seperti pada **Gambar 2.19**. Dengan ini didapatkan nilai  $T_U$  dan  $T_N$ .  $T_U$  merupakan waktu tunda sistem dan  $T_N$  merupakan waktu saat respon melewati waktu tunda hingga mencapai batas penarikan garis terhadap titik belok respon. Nilai  $T_U$  dan  $T_N$  berpengaruh terhadap orde *plant* dan untuk memodelkan respon *plant* berdasarkan tabel Strejc. Dengan diketahui nilai  $T_U$  dan  $T_N$ , maka dapat dicari nilai  $\tau$  berdasarkan **Persamaan (2.10)**. Metode Strejc melakukan pendekatan pada sistem yang tidak stabil tanpa *delay* (waktu tunda).

$$\tau = \frac{T_U}{T_N} \quad (2.10)$$



**Gambar 2.19** Respon *Step* Sistem dengan waktu  $t_1$ ,  $t_2$ ,  $t_b$ ,  $T_U$ , dan  $T_N$  [12]

1. Untuk  $\tau \geq 0$

Jika dari perbandingan didapatkan nilai  $\tau \geq 0$ , sistem didekati dengan orde ke-n dengan **Persamaan (2.11)** sebagai berikut:

$$G_{ST1}(s) = \frac{K}{(\tau_{ST} s + 1)^n} \quad (2.11)$$

Berdasarkan nilai  $\tau$  dapat ditentukan nilai  $y_i$  (dapat dilihat pada **Tabel 2.3**). Setelah didapatkan nilai  $y_i$ , maka melalui respon sistem dapat ditemukan nilai  $t_i$ . Jika nilai  $t_i$  sudah diketahui maka dapat ditemukan nilai  $\tau_{ST}$  seperti pada **Persamaan (2.12)**.

$$\tau_{ST} = \frac{t_i}{n-1} \quad (2.12)$$

**Tabel 2.3** Taksiran Orde ke-n dan Nilai Titik Singgung [12]

$n$	$\tau$	$y_i$	$n$	$\tau$	$y_i$
1	0,014	0,264	6	0,493	0,384
2	0,218	0,327	7	0,570	0,394
3	0,319	0,359	8	0,642	0,401
4	0,410	0,371	9	0,769	0,407
5	0,493	0,384	108	0,773	0,413

2. Untuk  $\tau < 0$

Jika dari perbandingan didapatkan nilai  $\tau < 0$ , sistem didekati dengan orde ke-n dengan **Persamaan (2.13)** sebagai berikut:

$$G_{ST2} = \frac{K}{(\tau_{ST1} s + 1)(\tau_{ST2} s + 1)} \quad (2.13)$$

Nilai  $\tau_{ST1}$  dan  $\tau_{ST2}$  dapat ditentukan dengan cara berikut:

- a. Mencari waktu  $t_1$  pada saat respon sistem mencapai 72% dari  $y_{ss}$ , kemudian tentukan jumlah dari  $\tau_{ST1}$  dan  $\tau_{ST2}$  berdasarkan **Persamaan (2.14)**.

$$\tau_{ST1} + \tau_{ST2} = \frac{t_1}{1,2564} \quad (2.14)$$

- b. Menentukan parameter  $t_2$  dari **Persamaan (2.15)**.

$$t_2 = 0,3574 (\tau_{ST1} + \tau_{ST2}) \quad (2.15)$$

- c. Menemukan nilai  $y(t_2)$  dari respon sistem berdasarkan **Gambar 2.19** dan menentukan perbandingan  $T$  untuk **Persamaan (2.16)** dari **Tabel 2.4**.

$$T = \frac{\tau_{ST1}}{\tau_{ST2}} \quad (2.16)$$

**Tabel 2.4** Nilai Konstan Perbandingan  $T$  [12]

$y(t_2)$	$T$	$y(t_2)$	$T$	$y(t_2)$	$T$
0,30	0,000	0,25	0,105	0,20	0,264
0,29	0,023	0,24	0,128	0,19	0,322
0,28	0,043	0,23	0,154	0,18	0,403
0,27	0,063	0,22	0,183	0,17	0,538
0,26	0,084	0,21	0,219	0,16	1,000

#### 2.4.2 Identifikasi Fisik [13]

Untuk identifikasi secara fisik, biasanya identifikasi model dibagi berdasarkan identifikasi mekanik atau elektrik.

- a. Identifikasi Fisik Sistem Mekanik

Identifikasi fisik pada sistem mekanik menganalisa model matematis berdasar komponen pendukungnya. Hukum dasar yang digunakan dalam identifikasi sistem mekanik adalah hukum Newton kedua yang hanya dapat diaplikasikan pada sistem mekanik. Dengan rumus  $F = m \cdot a$  dapat diartikan untuk perhitungan seluruh resultan gaya yang bekerja pada suatu sistem sama dengan turunan dari momentum linier benda tersebut terhadap waktu

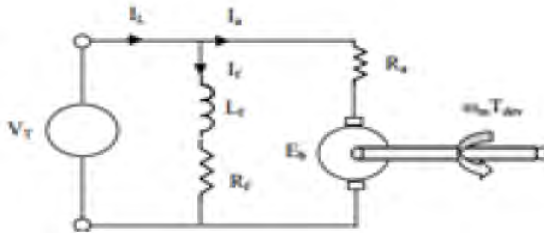
- b. Identifikasi Fisik Sistem Elektrik

Pada sistem elektrik, hukum dasar yang digunakan adalah hukum Kirchhoff arus dan tegangan. Hukum Kirchhoff arus (hukum *node/titik*) berbunyi bahwa penjumlahan seluruh arus yang masuk dan keluar adalah nol atau dapat dikatakan bahwa jumlah seluruh arus yang masuk dan seluruh arus yang keluar adalah sama. Hukum Kirchhoff tegangan (hukum *loop*) berbunyi bahwa penjumlahan

seluruh tegangan dalam satu *loop* pada sirkuit elektrik adalah nol atau dapat dikatakan bahwa jumlah seluruh tegangan naik sama dengan jumlah seluruh tegangan turun pada satu *loop* tertutup. Model matematis dari sebuah sirkuit elektrik dapat ditentukan dengan menggunakan salah satu atau kedua hukum Kirchhoff tersebut.

#### 2.4.2.1 Identifikasi Fisik Motor DC Konfigurasi Paralel

Konfigurasi motor DC paralel secara fisik dapat dilihat pada **Gambar 2.20**.



**Gambar 2.20** Konfigurasi Fisik Motor DC Konfigurasi Paralel

Motor DC dengan konfigurasi paralel memiliki karakteristik non linear terhadap arus jangkar pada **Persamaan (2.17)**, arus medan pada **Persamaan (2.18)**, dan torsi motor pada **Persamaan (2.19)**. Dari konfigurasi pada **Gambar 2.20** dapat dicari pemodelan fisik motor DC paralel sebagai berikut :

a. Rangkaian Arus Medan

$$I_f = \frac{1}{L_f s + R_f} E_a$$

$$L_f \dot{I}_f + R_f I_f = E_a$$

$$\dot{I}_f = -\frac{R_f}{L_f} I_f + \frac{1}{L_f} E_a \quad (2.17)$$

b. Rangkaian Arus Jangkar

$$I_a = \frac{1}{L_a s + R_a} (E_a - E_{agl})$$

$$L_a \dot{I}_a + R_a I_a = E_a - E_{ggl}$$

$$\dot{I}_a = -\frac{R_a}{L_a} I_a + \frac{1}{L_a} E_a - \frac{1}{L_a} E_{ggl}$$

$$E_{ggl} = K_{ggl} I_f \Omega_m, \text{ sehingga :}$$

$$\dot{I}_a = -\frac{R_a}{L_a} I_a + \frac{1}{L_a} E_a - \frac{1}{L_a} K_{ggl} I_f \Omega_m \quad (2.18)$$

c. Rangkaian Torsi Motor

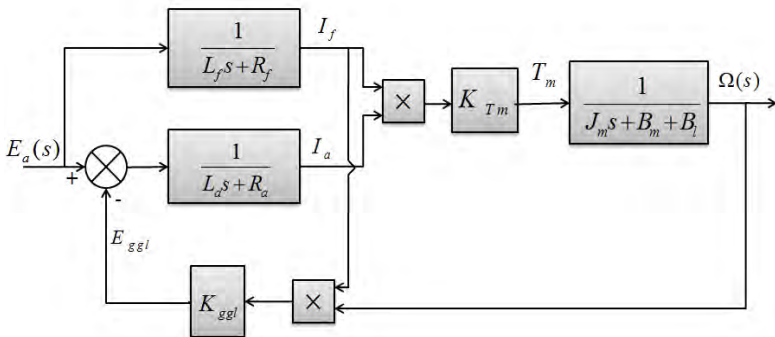
$$\Omega_m = \frac{(T_m - T_l)}{J_m s + B_m}$$

$$(J_m s + B_m) \Omega_m = T_m - (B_l \Omega_m)$$

$$J_m \dot{\Omega}_m + B_m \Omega_m = K_{Tm} I_a I_f - (B_l \Omega_m)$$

$$\dot{\Omega}_m = \frac{K_{Tm} I_a I_f}{J_m} - \frac{B_m \Omega_m}{J_m} - \frac{(B_l \Omega_m)}{J_m} \quad (2.19)$$

**Persamaan (2.17)** merupakan persamaan non linear motor DC terhadap arus medan, **Persamaan (2.18)** merupakan persamaan non linear motor DC terhadap arus jangkar, dan **Persamaan (2.19)** merupakan persamaan non linear motor DC terhadap torsi motor. Persamaan-persamaan diatas tersebut dapat direpresentasikan dalam diagram blok motor DC paralel seperti pada **Gambar 2.21**.



**Gambar 2. 21** Blok Diagram Fisik Motor DC Konfigurasi Paralel

## 2.5 Representasi *State Space* [13]

Dalam sistem pengaturan modern, sistem yang akan diatur menjadi lebih kompleks, terutama karena adanya persyaratan untuk mencapai akurasi yang tinggi meskipun sistem sangat kompleks. Sistem yang kompleks dapat memiliki lebih dari satu *input* dan *output*. Karena perlunya memenuhi persyaratan yang semakin ketat pada kinerja pengaturan sistem, peningkatan kompleksitas sistem dan kemudahan akses dalam skala besar, maka dikembangkan teori pengaturan modern yang merupakan pendekatan baru untuk analisis dan desain sistem kontrol yang kompleks. Teori modern ini dikembangkan sejak sekitar tahun 1960 dan didasarkan pada konsep *state*. Pendekatan baru yang kemudian disebut analisis *state space* ini kemudian menjadi teori baru yang marak digunakan pada sistem yang dinamik dan memiliki banyak kompleksitas.

Pada analisis *state space*, terdapat tiga variabel yang harus diperhatikan pada permodelan sistem dinamik yaitu variabel *input*, variabel *output* dan variabel *state*. Persamaan *state space* dapat diasumsikan pada sistem *Multiple Input Multiple Output* (MIMO) dengan  $n$  *integrator*. Dimana dapat diasumsikan terdapat  $r$  input yaitu  $u_1(t), u_2(t), \dots, u_r(t)$  dan  $m$  output yaitu  $y_1(t), y_2(t), \dots, y_m(t)$ . Tetapkan  $n$  *output* dari *integrator* sebagai variabel *state*  $x_1(t), x_2(t), \dots, x_n(t)$ , maka sistem dapat dituliskan pada **Persamaan (2.20)**.

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \dot{x}_2(t) &= f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)\end{aligned}\tag{2.20}$$

*output* sistem dapat dituliskan pada **Persamaan (2.21)**.

$$\begin{aligned}y_1(t) &= g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ y_2(t) &= g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ &\vdots \\ y_n(t) &= g_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)\end{aligned}\tag{2.21}$$

jika ditentukan,

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ \vdots \\ x_n(t) \end{bmatrix}, \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix},$$

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ \vdots \\ y_m(t) \end{bmatrix}, \mathbf{g}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix},$$

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ \vdots \\ u_r(t) \end{bmatrix}$$

Lalu **Persamaan (2.20)** dan **Persamaan (2.21)** dapat dituliskan sebagai :

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (2.22)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (2.23)$$

**Persamaan (2.22)** merupakan persamaan *state* dan **Persamaan (2.23)** merupakan persamaan *output*. Jika kedua persamaan tersebut dilinearisasikan dengan menggunakan operasi *state* dan vektor **f** dan **g** berdasar waktu, maka akan didapat persamaan linier dari persamaan *state* pada **Persamaan (2.24)** dan persamaan *output* pada **Persamaan (2.25)**.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.24)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (2.25)$$

Pada sistem dinamik dengan fungsi penghantar, representasi *state space* dapat dijelaskan dengan **Persamaan (2.26)**

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} \pm \dots + a_{n-1} s + a_n} \quad (2.26)$$

atau dapat dituliskan dalam bentuk persamaan diferensial pada **Persamaan (2.27)**,

$$y^{(n)} + a_1 y^{n-1} + \dots + a_n y = b_0 u^n + b_1 u^{n-1} + \dots + b_n u \quad (2.27)$$

lalu dengan memilih **Persamaan(2.28)**,

$$\begin{aligned} x_1 &= y - \beta_0 u \\ x_2 &= \dot{y} - \beta_0 \dot{u} - \beta_1 u = \dot{x}_1 - \beta_1 u \\ x_2 &= \dot{y} - \beta_0 \ddot{u} - \beta_1 \dot{u} - \beta_2 u = \dot{x}_2 - \beta_2 u \\ &\vdots \\ x_{n-1} &= y^{(n-2)} - \beta_0 u^{(n-2)} - \dots - \beta_{n-1} u = \dot{x}_n - \beta_{n-1} u \\ x_n &= y^{(n-1)} - \beta_0 u^{(n-1)} - \dots - \beta_{n-1} u = \dot{x}_{n-1} - \beta_{n-1} \end{aligned} \quad (2.28)$$

dengan  $\beta_0, \beta_1, \beta_2, \dots, \beta_n$  ditentukan melalui **Persamaan (2.29)** :

$$\begin{aligned} \beta_0 &= b_0 \\ \beta_1 &= b_1 - a_1 \beta_0 \\ \beta_2 &= b_2 - a_1 \beta_1 - a_2 \beta_0 \\ &\vdots \\ \beta_{n-1} &= b_{n-1} - a_1 \beta_{n-1} - a_{n-1} \beta_0 \\ \beta_n &= b_n - a_1 \beta_{n-1} - \dots - a_{n-1} \beta_1 - a_n \beta_0 \end{aligned} \quad (2.29)$$

maka didapatkan **Persamaan (2.30)** :

$$\begin{aligned} \dot{x}_1 &= x_2 + \beta_1 u \\ \dot{x}_2 &= x_3 + \beta_2 u \\ &\vdots \\ \dot{x}_{n-1} &= x_n + \beta_{n-1} u \\ \dot{x} &= -a_n x_1 - a_{n-1} x_2 - \dots - a_1 x_n + \beta_{n-1} u \end{aligned} \quad (2.30)$$



dan jika dilihat dalam bentuk persamaan vektor matriks, **Persamaan (2.30)** dapat dituliskan sebagai berikut :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \beta_0 u$$

atau :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (2.31)$$

$$y = \mathbf{C}\mathbf{x} + \mathbf{D}u \quad (2.32)$$

dengan:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \mathbf{D} = \beta_0 = b_0$$

Persamaan *state space* tidak harus mengambil variabel dari *output* sistem, tetapi dapat juga menggunakan variabel lain seperti *error* atau kombinasi linier lainnya.

## 2.6 Root Mean Square Error [2]

Untuk mengevaluasi terhadap perkiraan *error* digunakan komponen dari sebuah model statistika yang disebut RMSE. Carbone dan Armstrong (1982) menemukan bahwa *Root Mean Square Error* merupakan perhitungan perkiraan akurasi dari pemodelan sistem yang paling diminati. RMSE mengukur akurasi pada nilai deret waktu secara statistik seperti halnya regresi. RMSE dapat merepresentasikan ukuran dari *error* rata-rata karena RMSE membandingkan hasil data pengukuran dan data pemodelan pada skala yang sama antara kedua data tersebut. Formulasi perhitungan RMSE dapat dilihat pada **Persamaan (2.33)** dan formulasi nilai *error* dalam bentuk presentasi dapat dilihat pada **Persamaan (2.34)**.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}, \quad i = 1, 2, 3, \dots, n \quad (2.33)$$

$$e_i = \frac{A_i - M_i}{A_i} \times 100\%, \quad i = 1, 2, 3, \dots, n \quad (2.34)$$

dengan,

- n : Jumlah data
- i : Urutan data
- e : Nilai error
- A : Nilai data hasil pengukuran
- M : Nilai data hasil pemodelan

Pemodelan dengan menggunakan RMSE tidak dapat secara mutlak dikatakan terbaik atau sesuai dengan model dinamika sistem, namun untuk menentukan pemodelan yang mirip dengan data pengukuran dapat dilihat dengan nilai RMSE yang terkecil dari perhitungan lainnya.

## 2.7 Kontroler [14]

Dalam sebuah sistem kontrol, kontroler mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem, yaitu kontroler. Kontroler merupakan salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*set point*) atau sinyal *error*

mejadi sinyal kontrol. Sinyal *error* disini adalah selisih antara sinyal umpan balik yang dapat berupa sinyal keluaran *plant* sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (*set point*). Salah satu tugas komponen kontroler adalah meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal *set point* dan sinyal aktual. Hal ini sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang senantiasa (diinginkan) sama dengan sinyal *set point*. Semakin cepat reaksi sistem (sinyal aktual) mengikuti sinyal *set point* dan semakin kecil kesalahan yang terjadi, semakin baiklah kinerja sistem kontrol yang diterapkan.

### 2.7.1 Kontroler *Proportional Integral Derivative* (PID) [14]

Salah satu jenis kontroler berdasarkan aksi kontrolnya adalah kontroler PID. Kontroler PID merupakan kontroler yang aksi kontrolnya merupakan kombinasi dari aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol differensial. Untuk kontroler PID, sinyal kesalahan  $e(t)$  merupakan masukan kontroler sedangkan keluaran kontroler adalah sinyal kontrol  $u(t)$ . Hubungan antara masukan kontroler  $e(t)$  dan keluaran kontroler  $u(t)$  dapat dilihat pada **Persamaan (2.35)**.

$$u(t) = K_p \left\{ e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de(t)}{dt} \right\} \quad (2.35)$$

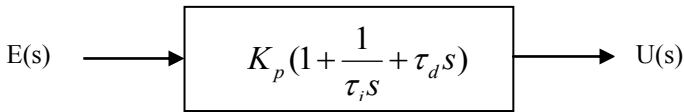
atau dalam besaran transformasi Laplace :

$$U(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s) \quad (2.36)$$

$K_p$  adalah penguatan proporsional,  $\tau_i$  adalah waktu integral dan  $\tau_d$  adalah waktu differensial. Parameter  $K_p$ ,  $\tau_i$ , dan  $\tau_d$  ketiganya dapat ditentukan. Sehingga fungsi alih kontroler PID seperti pada **Persamaan (2.37)**.

$$\frac{U(s)}{E(s)} = K_p \left\{ 1 + \frac{1}{\tau_i s} + \tau_d s \right\} \quad (2.37)$$

Diagram blok kontroler PID adalah seperti pada **Gambar 2.22** sebagai berikut :



**Gambar 2.22** Diagram Blok Kontroler PID

Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi dari ketiga parameter  $P$ ,  $I$  dan  $D$ . Pengaturan konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diatur lebih menonjol dibanding yang lain. Pengaruh parameter  $K_p$ ,  $K_i$ , dan  $K_d$  terhadap respon dapat dilihat pada **Tabel 2.5**.

**Tabel 2.5** Pengaruh Parameter  $K_p$ ,  $K_i$ , dan  $K_d$

<i>Closed-Loop Response</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error Steady State</i>
$K_p$	<i>Decrease</i>	<i>Increase</i>	<i>Small Change</i>	<i>Decrease</i>
$K_i = K_p \cdot \frac{1}{\tau_i}$	<i>Decrease</i>	<i>Increase</i>	<i>Increase</i>	<i>Eliminate</i>
$K_d = K_p \cdot \tau_d$	<i>Small Change</i>	<i>Decrease</i>	<i>Decrease</i>	<i>Small Change</i>

### 2.7.2 Kontroler Optimal *Linear Quadratic Regulator* (LQR) [15] [16]

*Linear Quadratic Regulator* adalah suatu kontrol optimal pada sistem linear dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator (*Regulator Problem*). Dalam pengaturan optimal ada 3 hal yang dapat dijadikan acuan untuk merancang kontroler, yaitu :

1. Meminimumkan waktu
2. Meminimumkan kesalahan (*error*)
3. Meminimumkan energi kontrol

Pada Tugas Akhir ini dipilih metode pengaturan optimal yang dapat meminimumkan kesalahan (*error*) ketika terjadi pembebanan.

Desain kontrol optimal dilakukan melalui formulasi variabel *state*. Pada dasarnya solusi kontrol optimal dititikberatkan pada pencarian nilai sinyal kontrol optimal  $\mathbf{u}(t)$ , sehingga indeks performansi dioptimasi.

Suatu *plant* linear dalam *state space* pada **Persamaan (2.38)** dinyatakan sebagai berikut:

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (2.38)$$

dengan,

$\mathbf{A}$  : nxn matriks *state*  
 $\mathbf{B}$  : nxr matriks *control*  
 $\mathbf{x}(t)$  : nth vektor *state*  
 $\mathbf{u}(t)$  : rth vektor *control*

Adapun indeks performansi dalam interval  $[0, \infty]$  secara matematis dinyatakan pada **Persamaan (2.39)** sebagai berikut :

$$J = \int_0^{\infty} [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)]dt \quad (2.39)$$

dengan,

$t_0$  : waktu awal  
 $\infty$  : waktu akhir (*infinite*)  
 $\mathbf{Q}$  : matriks nxn simetri, semidefinit positif  
 $\mathbf{R}$  : matriks mxm simetri, definit positif

Metode kontrol optimal bertujuan untuk menjaga agar *state*  $\mathbf{x}(t)$  mendekati 0 yang disebut *state regulator system*. Dengan kata lain, untuk mendapatkan sinyal kontrol  $\mathbf{u}(t)$ , apabila harga keluaran awal tidak nol, maka masukannya harus ditentukan agar keluarannya menjadi nol.

Kontrol optimal diberikan sebagai berikut :

$$\mathbf{u}^*(t) = -\mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}}\mathbf{x}^*(t) = -\mathbf{K}\mathbf{x}^*(t) \quad (2.40)$$

dengan Kalman *gain*,

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}} \quad (2.41)$$

$\bar{\mathbf{P}}$ ,  $n \times n$  simetri, definit positif, matriks konstan, dan merupakan solusi dari nonlinier dengan persamaan *Algebraic Riccati Equation* (ARE) yang dituliskan sebagai berikut :

$$\mathbf{A}^T \bar{\mathbf{P}} + \bar{\mathbf{P}} \mathbf{A} - \bar{\mathbf{P}} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} + \mathbf{Q} = 0 \quad (2.42)$$

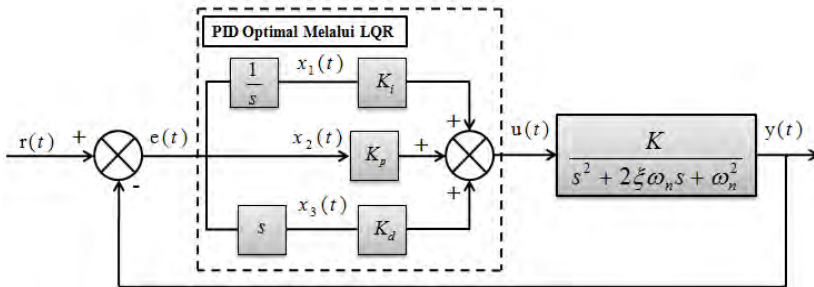
Penentuan kontrol objektif pada sistem kontrol optimal dan menentukan matriks pembobotan  $\mathbf{Q}$  dan  $\mathbf{R}$ , belum ada solusi yang unik dari matriks-matriks tersebut. Untuk menentukan matriks pembobotan tersebut biasanya dilakukan dengan *trial and error* (coba-coba) di mana pemilihan pemberat  $\mathbf{Q}$  dan  $\mathbf{R}$  berpedoman pada:

- Semakin besar harga  $\mathbf{Q}$ , semakin memperbesar harga elemen penguatan  $\mathbf{K}$  sehingga mempercepat sistem untuk mencapai *steady state*.
- Semakin besar harga  $\mathbf{R}$ , maka akan memperkecil harga penguatan  $\mathbf{K}$  dan memperlambat sistem mencapai *steady state*.

### 2.7.3 Kontroler PID-LQR [17] [18]

Masalah utama kontroler PID adalah *tuning*, yaitu menentukan nilai parameter  $K_p$ ,  $K_i$ , dan  $K_d$  agar diperoleh performansi sistem yang optimal. Umumnya proses *tuning* parameter PID masih dilakukan secara manual, sehingga memakan waktu. Yu mengembangkan metode LQR (*Linear Quadratic Regulator*) untuk *tuning* PID dengan hasil yang optimal, tetapi perlu kalkulasi matematika dan penyelesaian yang rumit.

Metode PID-LQR merupakan metode PID optimal dengan pendekatan LQR untuk menentukan parameter  $K_p$ ,  $K_i$  dan  $K_d$  pada kontroler PID. Diagram blok PID-LQR dapat ditunjukkan pada **Gambar 2.23**.



**Gambar 2.23** Diagram Blok PID-LQR

Pada metode PID-LQR terdapat beberapa prosedur yang dilakukan untuk mendapatkan parameter kontroler PID. Langkah-langkah perancangan sistem kontrol PID-LQR dapat dijelaskan sebagai berikut :

1. Fungsi alih yang didapatkan dari proses identifikasi *plant* harus diubah dalam representasi *state space*. Diketahui bahwa *plant* merupakan orde dua dengan *input*  $u(t)$ , *output*  $y(t)$ , dan fungsi alih *plant* seperti pada **Persamaan (2.43)**.

$$\frac{Y(s)}{U(s)} = \frac{a}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{-E(s)}{U(s)} \quad (2.43)$$

Selanjutnya dimisalkan sebuah *state* :

$$x_0 = \int e(t)dt, \quad x_1 = e(t), \quad x_2 = \frac{de(t)}{dt} \quad (2.44)$$

Representasi *plant* dalam *state space* dapat dibentuk sebagai berikut:

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} u \quad (2.45)$$

sehingga diperoleh parameter *plant* sebagai berikut :

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_n^2 & -2\xi\omega_n \end{bmatrix} ; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} \quad (2.46)$$

2. Menentukan matriks pembobot **Q** dan **R**  
Matriks **Q** merupakan koefisien pembobot yang digunakan untuk menentukan lebar area *state*, sedangkan matriks **R** digunakan untuk menentukan lebar area sinyal kontrol **u**.
3. Menyelesaikan persamaan *Algebraic Riccati Equation* (ARE).  
Persamaan *Algebraic Riccati Equation* (ARE) seperti ditunjukkan pada **Persamaan (2.47)** :

$$\mathbf{A}^T \bar{\mathbf{P}} + \bar{\mathbf{P}} \mathbf{A} - \bar{\mathbf{P}} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} + \mathbf{Q} = 0 \quad (2.47)$$

Penyelesaian ini digunakan untuk memperoleh matriks Riccati  $\mathbf{P}$  dengan bantuan matriks  $\mathbf{A}$ , matriks  $\mathbf{B}$ , dan matriks pembobot  $\mathbf{Q}$  dan  $\mathbf{R}$ .

4. Mendapatkan parameter PID berdasarkan persamaan Kalman *gain*. Persamaan Kalman *gain* ditunjukkan pada **Persamaan (2,48)** sebagai berikut :

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} \quad (2.48)$$

$$\mathbf{K} = \mathbf{R}^{-1} \begin{bmatrix} 0 & 0 & -a \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} \quad (2.49)$$

$$\mathbf{K} = -\mathbf{R}^{-1} \begin{bmatrix} a \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} \quad (2.50)$$

$$\mathbf{K} = -[a\mathbf{R}^{-1}P_{13} : a\mathbf{R}^{-1}P_{23} : a\mathbf{R}^{-1}P_{33}] \quad (2.51)$$

$$\mathbf{K} = -[K_i : K_p : K_d] \quad (2.52)$$

Kemudian sinyal kontrol optimal solusi PID-*Linear Quadratic Regulator* dapat direpresentasikan ke dalam **Persamaan (2.53)** sebagai berikut :

$$\mathbf{u}^*(t) = -\mathbf{K}\mathbf{x}^*(t) \quad (2.53)$$

$$\mathbf{u}^*(t) = K_i x_1(t) + K_p x_2(t) + K_d x_3(t) \quad (2.54)$$

$$\mathbf{u}^*(t) = K_i \int e(t)dt + K_p e(t) + K_d \frac{de(t)}{dt} \quad (2.55)$$



## BAB 3

### PERANCANGAN SISTEM

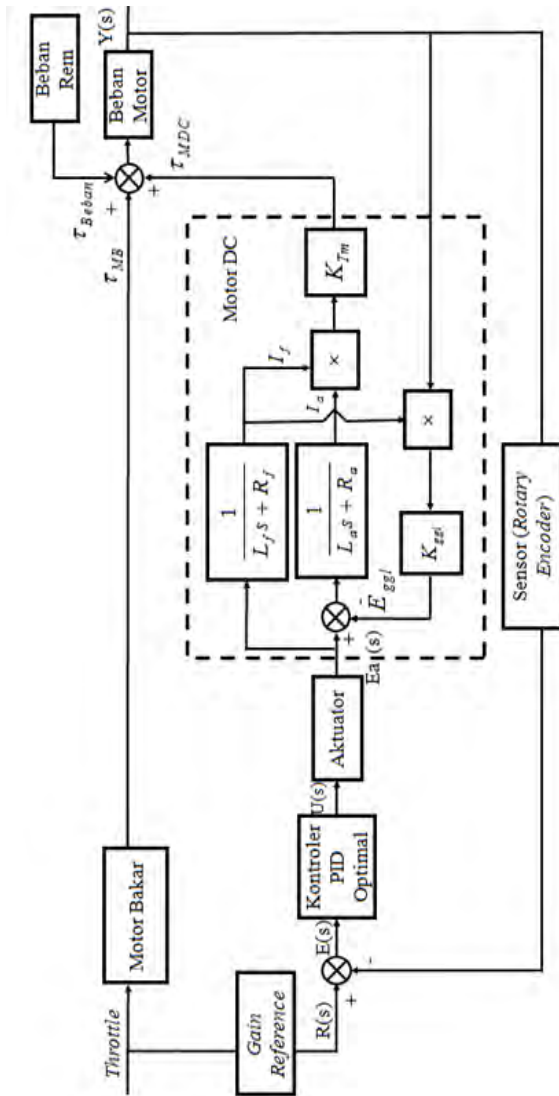
#### 3.1 Gambaran Umum Sistem

HEV merupakan kendaraan listrik hibrid yang bekerja dengan mengkombinasikan pemanfaatan mesin bakar dan motor listrik sebagai tenaga penggerak. HEV memiliki beberapa konfigurasi yaitu konfigurasi seri, konfigurasi paralel, dan konfigurasi seri-paralel. Pada Tugas Akhir ini digunakan Simulator *Hyrid Electric Vehicle* dengan konfigurasi paralel. Prinsip kerja HEV dengan konfigurasi paralel adalah mesin bakar dan motor listrik dapat menjalankan kendaraan secara bersama-sama dengan mesin bakar sebagai penggerak utama dan motor DC sebagai penggerak pembantu. Pada konfigurasi paralel peran motor DC sangat penting untuk membantu mesin bakar. Ketika mesin bakar sebagai penggerak utama diberikan efek beban yang berlebih, maka kecepatan putar yang dihasilkan mesin bakar akan berkurang. Sebagai penggerak pembantu, Motor DC akan memberikan bantuan sehingga sesuai dengan kecepatan putar yang diinginkan. Pengujian ini dilakukan dengan menggunakan rem elektromagnetik sebagai representasi efek beban untuk diberikan kepada mesin bakar ketika telah mencapai kecepatan stabil. Dengan demikian dapat diketahui fungsi motor DC untuk membantu kerja dari mesin bakar.

Simulator HEV yang digunakan pada Tugas Akhir ini merupakan hasil rancangan dari Wahyu Sriwidodo dan Galih Satriyo A.W. pada Tugas Akhir pada tahun 2010 seperti terlihat pada **Gambar 3.1**. Diagram blok sistem dapat dilihat pada **Gambar 3.2**.



**Gambar 3.1** Simulator HEV dengan Konfigurasi Paralel

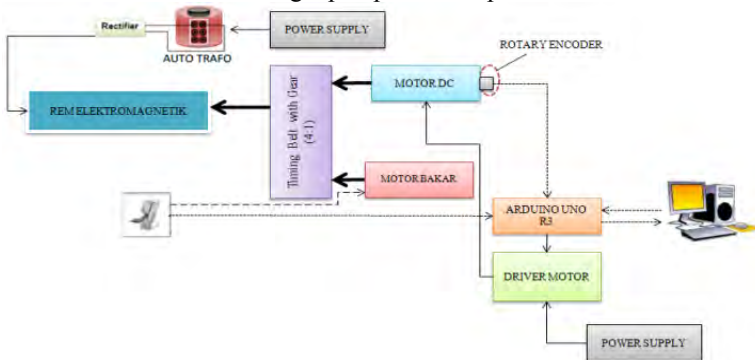


Gambar 3.2 Diagram Blok Sistem

Pada Tugas Akhir ini digunakan metode kontrol berbasis kontrol optimal yaitu *PID-Linear Quadratic Regulator*. Metode kontrol tersebut digunakan untuk mengatur sinyal kontrol yang diperlukan untuk mengatur kecepatan motor DC sehingga ketika diberi beban berlebih, kecepatan putar sesuai dengan yang diharapkan.

### 3.2 Perancangan Perangkat Keras

Secara garis besar simulator PHEV merupakan gabungan antara ICE dan motor DC, selain itu juga terdapat perangkat pendukung lainnya yang bekerja secara sinergi. Konfigurasi perangkat keras simulator PHEV secara lengkap dapat dilihat pada **Gambar 3.3**.



**Gambar 3.3** Konfigurasi Perangkat Keras Simulator PHEV

Pada konfigurasi HEV secara lengkap terdapat motor DC, mesin bakar, rem elektromagnetik, sistem minimum Arduino, dan *rotary encoder*. Selain itu, untuk mengubah tegangan AC menjadi tegangan DC diperlukan rangkaian penyearah gelombang penuh sebagai tegangan masukan motor DC. Dilihat dari spesifikasi, putaran dari mesin bakar jauh lebih besar dari motor DC sehingga kecepatan putar ICE harus dikurangi secara proporsional. Oleh karena itu diperlukan transmisi gir agar kecepatan putar pada ICE sebanding dengan motor DC.

Potensiometer juga diperlukan pada perancangan dan dipasang pada pedal gas untuk merepresentasikan bukaan *throttle* yang merupakan masukan pada ICE, terletak pada karburator dan terhubung dengan pedal gas. Sedangkan keluaran sistem diukur menggunakan *rotary encoder*. Keluaran pada sistem merupakan kecepatan putar pada poros motor DC sebagai poros utama yang akan menanggung beban

berupa rem elektromagnetik yang menghasilkan torsi lawan terhadap mesin bakar. Selain itu juga diperlukan sistem minimum Arduino dan rangkaian *driver* untuk sistem pengaturannya.

Perancangan perangkat keras untuk simulator HEV dalam Tugas Akhir ini dapat dibagi menjadi 2 bagian yaitu perangkat mekanik dan perangkat elektronik

### 3.2.1 Perancangan Perangkat Mekanik

Terdapat beberapa perangkat dalam perancangan mekanik untuk memenuhi suatu sistem seperti konfigurasi ICE dan motor DC pada bagian perangkat utama dan juga perangkat pendukung lainnya.

#### 3.2.1.1 *Internal Combustion Engine (ICE)*

Pada simulator HEV yang digunakan dalam Tugas Akhir ini digunakan mesin pemotong rumput. Mesin pemotong rumput direpresentasikan sebagai ICE yang berperan sebagai penggerak utama untuk menanggung beban. Mesin pemotong rumput yang digunakan adalah jenis mesin bakar 2 Tak. Mesin pemotong rumput 2 tak dipilih karena memiliki kecepatan putar yang tinggi dan poros engkol terhubung secara langsung dengan poros keluaran tanpa ada sistem transmisi. Spesifikasi mesin pemotong rumput 2 Tak dapat dilihat pada **Tabel 3.1**. Bentuk fisik mesin pemotong rumput 2 Tak yang digunakan pada simulator HEV dapat dilihat pada **Gambar 3.4**.

**Tabel 3.1** Spesifikasi Mesin Pemotong Rumput 2 Tak

No	Klasifikasi	Spesifikasi
1	Jenis	2 Tak
2	Bahan Bakar	Bensin dan Oli Samping
3	Perbandingan Bensin dan Oli	25 : 1
4	Kapasitas Silinder	33.8 cc
5	Pemasukan Bahan Bakar	Karburator
6	Starter	Recoil Starter
7	Sistem Pengapian	Electronic Ignition
8	Sistem Kopling	Kering, Sentrifugal
9	Kecepatan Putar Maks	8000 rpm
10	Daya Keluaran	1.8 HP



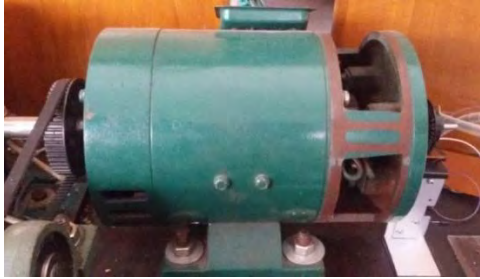
**Gambar 3.4** Bentuk Fisik Mesin Pemotong Rumput 2 Tak pada Simulator HEV

### 3.2.1.2 *Motor Listrik*

Pada simulator HEV digunakan motor listrik jenis motor DC. Motor DC merupakan motor yang mengubah energi listrik DC menjadi energi gerak. Pada simulator HEV, motor DC berfungsi sebagai penggerak pembantu, oleh karena itu dipilih spesifikasi motor DC yang memiliki daya keluaran lebih kecil daripada ICE. Selain itu, Motor DC dipilih karena motor DC lebih mudah dikendalikan dibandingkan dengan motor *Alternating Current* (AC). Pada Tugas Akhir ini, motor DC akan diatur agar dapat membantu kerja ICE ketika menanggung beban. Spesifikasi motor DC yang digunakan dapat dilihat pada **Tabel 3.2**. Bentuk fisik motor DC yang digunakan pada simulator HEV dapat dilihat pada **Gambar 3.5**.

**Tabel 3.2** Spesifikasi Motor DC

No	Klasifikasi	Spesifikasi
1	Tegangan Maksimal	190 V
2	Kecepatan Putar Maks	1750 rpm
3	Tipe	Seri/Paralel
4	Daya Keluaran	0.5 HP



**Gambar 3.5** Bentuk Fisik Motor DC pada Simulator HEV

Motor DC yang digunakan merupakan motor DC universal, lalu pada simulator HEV motor DC dikonfigurasi secara paralel.

### 3.2.1.3 Roda Gigi Penghubung

Pada simulator HEV ini, poros ICE dan poros motor DC sebagai poros utama yang terhubung dengan beban terpasang secara terpisah. Untuk menyatukan poros ICE dan poros motor DC dibutuhkan pemasangan *timing belt* dan *gear*. Dengan demikian, ICE dapat menyalurkan daya ke poros utama. Sebagaimana diketahui bahwa putaran ICE lebih besar daripada putaran motor DC, sehingga perlu adanya transmisi *gear* yang dapat mereduksi kecepatan ICE secara proporsional sehingga sesuai dengan rentang kecepatan putar motor DC.

Perbandingan roda gigi digunakan untuk menyamakan rentang kecepatan putar antara keduanya. Rentang kerja ICE antara 0 – 8000 rpm dan rentang kerja motor DC pada 0 – 2000 rpm, dengan menggunakan teori perbandingan roda gigi dapat dicari perbandingan *gear* yang harus digunakan dengan rumus seperti pada **Persamaan (3.1)**.

$$\omega_a R_a = \omega_b R_b \quad (3.1)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_a} \quad (3.2)$$

$$\frac{2000}{8000} = \frac{R_b}{R_a}$$

$$\frac{R_a}{R_b} = \frac{4}{1} \quad (3.3)$$

dengan,

- $\omega_a$  : Kecepatan putar di motor DC
- $\omega_b$  : Kecepatan putar di ICE
- $R_a$  : Jari-jari gir pada poros motor DC
- $R_b$  : Jari-jari gir pada poros ICE

Berdasarkan persamaan didapatkan perbandingan roda gigi 1: 4 untuk menyamakan kecepatan putar antara ICE dan motor DC. Akan tetapi, secara fisik perbandingan roda gigi 1:4 tidak efisien dari segi tempat. Untuk efisiensi tempat, maka perbandingan *gear* dibagi menjadi dua bagian dengan memberikan *gear* tambahan di antara keduanya seperti terlihat pada **Gambar 3.6**.



**Gambar 3.6** Transmisi *Gear*

Perbandingan  $R_{c1} : R_{c2} = 2:1$ , *gear* poros ICE  $R_b$  digandeng dengan *gear* poros tengah  $R_{c1}$  dengan perbandingan  $R_b : R_{c1} = 1:2$ . *Gear* poros tengah  $R_{c2}$  digandeng dengan *gear* poros motor DC  $R_a$  dengan perbandingan  $R_{c2} : R_a = 1:2$ , sehingga perhitungan perbandingan *gear* yang digunakan pada simulator HEV adalah seperti pada **Persamaan (3.4)** sebagai berikut :

$$\omega_a R_a = \omega_{c2} R_{c2}$$

$$\omega_{c2} = \frac{\omega_a R_a}{R_{c2}} \quad (3.4)$$

$$\omega_b R_b = \omega_{c1} R_{c1}$$

$$\omega_{c1} = \frac{\omega_b R_b}{R_{c1}} \quad (3.5)$$

dengan,

$\omega_{c1}$  : Kecepatan putar pada poros tengah 1

$\omega_{c2}$  : Kecepatan putar pada poros tengah 2

$R_{c1}$  : Jari-jari gir pada poros tengah 1

$R_{c2}$  : Jari-jari gir pada poros tengah 2

karena  $\omega_{c1} = \omega_{c2}$ , maka :

$$\frac{\omega_a R_a}{R_{c2}} = \frac{\omega_b R_b}{R_{c1}} \quad (3.6)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_{c1}} \cdot \frac{R_{c2}}{R_a}$$

$$\frac{\omega_a}{\omega_b} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \quad (3.7)$$

Berdasarkan pada persamaan, dapat dilihat bahwa apabila  $\omega_a$  diberikan nilai 2000 rpm maka akan didapatkan nilai  $\omega_b$  sebesar 8000 rpm. Sehingga dengan konfigurasi penyatuan poros akan menghasilkan nilai reduksi yang sama dengan menggunakan perbandingan *gear* secara langsung.

Selain kecepatan putar, torsi juga akan mengalami perubahan dengan adanya reduksi ini. Hubungan antara perbandingan torsi dengan jari-jari *gear* dapat dilihat pada **Persamaan (3.8)**.

$$\frac{\tau_a}{R_a} = \frac{\tau_b}{R_b} \quad (3.8)$$

$$\frac{\tau_a}{\tau_b} = \frac{\omega_a}{\omega_b} \quad (3.9)$$

$$\frac{\tau_a}{\tau_b} = \frac{8000}{2000} \quad (3.10)$$

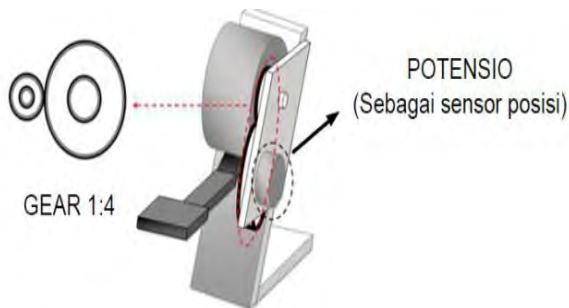
$$\tau_a = 4 \tau_b \quad (3.11)$$



Dapat disimpulkan dari persamaan bahwa poros utama akan memiliki torsi sebesar empat kali torsi yang dihasilkan ICE.

#### 3.2.1.4 Sensor Posisi

Pada Simulator HEV, *set point* diberikan melalui pedal gas yang merupakan aktuator untuk mengatur lebar bukaan *throttle* pada karburator yang terhubung langsung dengan klep karburator melalui kabel. Pijakan pedal gas sebanding dengan bukaan *throttle* karburator. Oleh karena itu, untuk mengetahui bukaan *throttle* pada karburator dapat dilihat melalui pemberian sensor posisi berupa potensiometer pada pedal gas. Karena penempatan sensor posisi tidak dimungkinkan berada pada karburator, maka sensor posisi diletakkan pada pedal. Oleh karena itu derajat putaran pedal gas merepresentasikan bukaan klep pada karburator. Bentuk fisik sensor posisi pada HEV dapat dilihat pada **Gambar 3.8**.



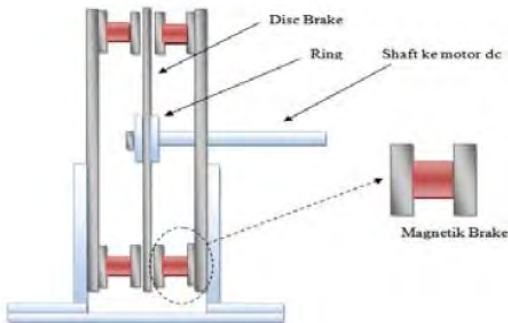
**Gambar 3.7** Ilustrasi Sensor Posisi pada Pedal Gas



**Gambar 3.8** Potensiometer pada Simulator HEV

### 3.2.1.5 Rem Elektromagnetik

Rem elektromagnetik pada simulator HEV ini berfungsi sebagai beban. Rem elektromagnetik diletakkan pada poros motor DC. Rem elektromagnet ini terdiri dari dua piringan besi yang diapit oleh dua penghasil medan magnet yang terdiri dari beberapa magnetik *brake* di sisi kiri dan kanan piringan. Magnetik *brake* ini terbuat dari konduktor yang telah dililit kumparan, karena dialiri arus listrik, maka akan timbul medan magnet. Konfigurasi dari rem elektromagnet ini dapat dilihat pada **Gambar 3.9**.



**Gambar 3.9** Konfigurasi Rem Elektromagnetik

Prinsip kerja dari rem elektromagnetik ini adalah ketika kecepatan poros utama tinggi, medan magnet yang dihasilkan oleh magnetik *brake* akan menimbulkan arus *eddy*, sehingga menyebabkan gaya lawan yang berlawanan dengan arah putar dari piringan besi. Hal ini akan menyebabkan terjadinya penurunan kecepatan putar pada motor DC. Gaya lawan inilah yang disebut sebagai beban bagi simulator HEV. Bentuk fisik rem elektromagnetik dapat dilihat pada **Gambar 3.10**.



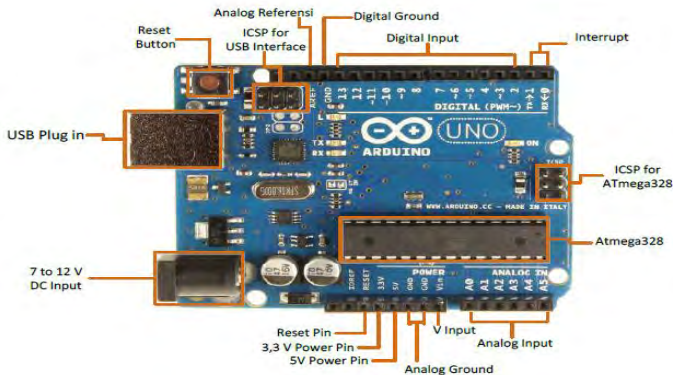
**Gambar 3.10** Bentuk Fisik Rem Elektromagnetik

### 3.2.2 Perancangan Perangkat Elektronik

Perancangan elektronik dibutuhkan pada simulator HEV berupa rangkaian elektronik untuk mendukung sistem seperti Arduino, *rectifier*, dan *rotary encoder*.

#### 3.2.2.1 Sistem Minimum Arduino

Untuk keperluan akuisisi data dan pengolahan data pada Tugas Akhir ini digunakan sistem minimum Arduino R3 berbasis ATmega 328. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 analog *input*, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Pengiriman data dari sistem minimum Arduino ke PC/*laptop* dilakukan melalui USB. Bahasa yang digunakan oleh perangkat Arduino ini yaitu menggunakan bahasa C+ untuk pemrogramannya. *Software* Arduino ini dilengkapi dengan kumpulan *library* yang cukup lengkap, sehingga dapat membantu pengguna dalam penggunaannya. Sistem minimum Arduino Uno R3 secara lengkap dapat dilihat pada **Gambar 3.11**.



**Gambar 3.11** Bentuk Fisik Sistem Minimum Arduino Uno R3

#### 3.2.2.2 Driver Motor DC

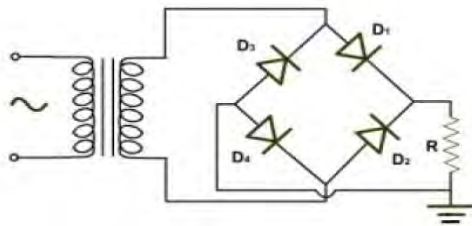
*Driver* motor DC merupakan rangkaian yang digunakan untuk memberikan tegangan *input* pada motor DC. *Driver* berfungsi sebagai aktuator yang mengubah besaran sinyal kontrol menjadi besaran yang digunakan untuk mengoperasikan motor DC tersebut. Rangkaian *driver* motor DC memanfaatkan sinyal PWM yang dihasilkan oleh IC regulator dan memanfaatkan cara kerja dua transistor dimana arus yang mengalir

### 3.2.2.3 Rangkaian Penyearah Gelombang Penuh

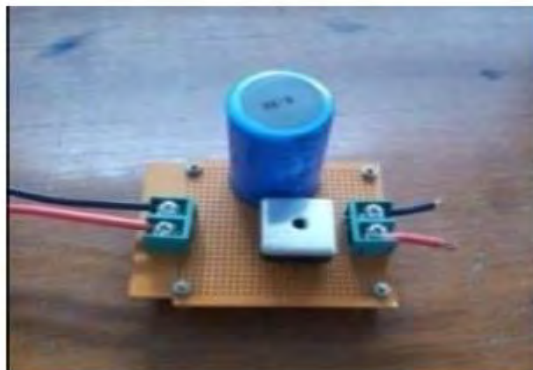
52

gelombang ini memiliki komponen utama yaitu dioda. Dioda memiliki karakteristik *short circuit* ketika *forward bias* dan *open circuit* ketika *reverse bias*. Skema rangkaian penyearah gelombang penuh dapat dilihat pada **Gambar 3.14**. Bentuk fisik rangkaian penyearah gelombang penuh dapat dilihat pada **Gambar 3.15**.

Ketika tegangan masukan bernilai positif, maka dioda D1 dan D4 berada pada posisi *forward bias* sehingga tegangan keluaran pada R bernilai positif. Ketika tegangan masukan bernilai negatif, maka dioda D2 dan D3 berada pada posisi *forward bias* sehingga tegangan keluaran pada R akan bernilai positif. Hasil dari proses ini tidak didapatkan tegangan DC murni melainkan tegangan DC yang berubah secara periodik dan masih mengandung komponen tegangan bolak – balik. Variasi tegangan ini disebut dengan tegangan riak, sehingga untuk memperkecil riak tegangan dapat menggunakan filter kapasitor.



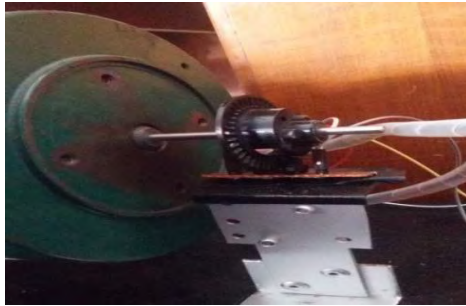
**Gambar 3.14** Skema Rangkaian Penyearah Gelombang Penuh



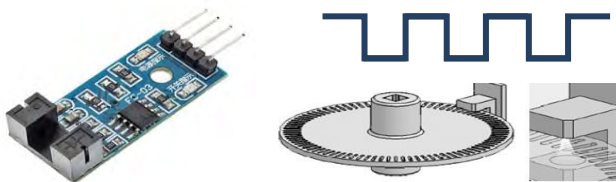
**Gambar 3.15** Bentuk Fisik Rangkaian Penyearah Gelombang Penuh pada Simulator HEV

#### 3.2.2.4 Rotary Encoder

*Rotary encoder* digunakan sebagai sensor kecepatan pada HEV. Pada *rotary encoder* terdapat *optocoupler* ini dipasang pada *shaft* motor DC seperti pada **Gambar 3.16**, sehingga kecepatan motor DC bisa terbaca. *Rotary encoder* akan mendeteksi tiap lubang pada *encoder disk* dan mengubahnya dalam bentuk pulsa digital 0 1. *Rotary encoder* mendeteksi lubang pada *encoder disk* dengan menggunakan *optocoupler* yang mentransmisikan cahaya diantara celah *disk*, bila cahaya melewati lubang *disk* dan diterima *receiver*, maka sensor akan mengeluarkan pulsa 1, sedangkan saat cahaya tidak melewati lubang *disk*, maka sensor akan mengeluarkan pulsa 0. Modul *optocoupler* dan *encoder disk* dapat dilihat pada **Gambar 3.17**.

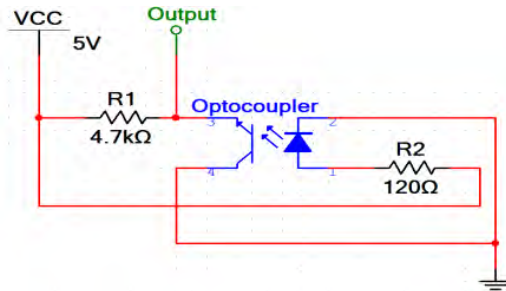


**Gambar 3.16** *Rotary Encoder* pada Simulator HEV



**Gambar 3.17** Modul *Optocoupler* dan *Encoder Disk*

Rangkaian *optocoupler* dibuat agar *optocoupler* bisa bekerja dengan baik dalam pengambilan data kecepatan motor DC. Pembuatan rangkaian tersebut berdasarkan data dari *datasheet* yang dimiliki *optocoupler*. Skema gambar rangkaian *optocoupler* ditunjukkan **Gambar 3.18**.



**Gambar 3.18** Skema Rangkaian *Optocoupler*

### 3.3 Perancangan Perangkat Lunak

Perangkat lunak diperlukan dalam perancangan sistem sebagai *interface* antara *plant* dan komputer. Perangkat lunak yang digunakan dalam pengerjaan Tugas Akhir ini antara lain adalah penggunaan perangkat lunak Arduino, MATLAB, dan LabView. Perangkat lunak digunakan untuk pendukung penggunaan perangkat keras, pengambilan data identifikasi *plant*, dan simulasi kontroler pada *plant*.

#### 3.3.1 *Software* Arduino

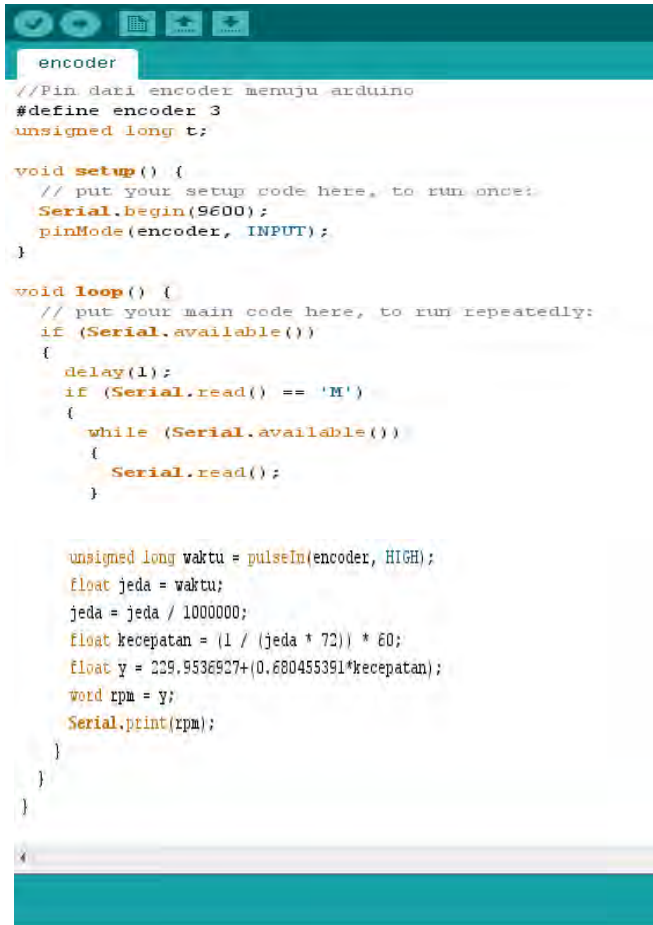
Penggunaan sistem Arduino pada Tugas Akhir ini membutuhkan perancangan program untuk menjalankan fungsi dari minimum sistem tersebut. Pada Arduino, program dituliskan pada *software* Arduino yang telah disediakan pada *Arduino.cc*, lalu program dituliskan pada Arduino *board* melalui serial USB yang dihubungkan pada komputer. Dalam Tugas Akhir ini, program Arduino digunakan untuk membaca *input analog* dari *output rotary encoder* untuk mengetahui respon kecepatan putar HEV. Tampilan program Arduino yang dirancang dapat dilihat pada **Gambar 3.19**.

Pada program Arduino yang dibuat dilakukan linearisasi antara kecepatan motor DC hasil pembacaan *rotary encoder* dengan kecepatan motor DC hasil pembacaan *tachometer digital* yang ditunjukkan pada **Persamaan (3.12)**. Hal ini dilakukan agar data kecepatan yang diolah oleh LabView hanya memiliki sedikit perbedaan dengan data kecepatan asli.

$$f(x) = 229,9536927 + 0,680455391x \quad (3.12)$$

dengan,

$f(x)$  : kecepatan *output* Arduino  
 $x$  : kecepatan *output* rotary encoder



```

encoder

//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(encoder, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    delay(1);
    if (Serial.read() == 'M')
    {
      while (Serial.available())
      {
        Serial.read();
      }

      unsigned long waktu = pulseIn(encoder, HIGH);
      float jeda = waktu;
      jeda = jeda / 1000000;
      float kecepatan = (1 / (jeda * 72)) * 60;
      float y = 229.9536927+(0.680455391*kecepatan);
      word rpm = y;
      Serial.print(rpm);
    }
  }
}

```

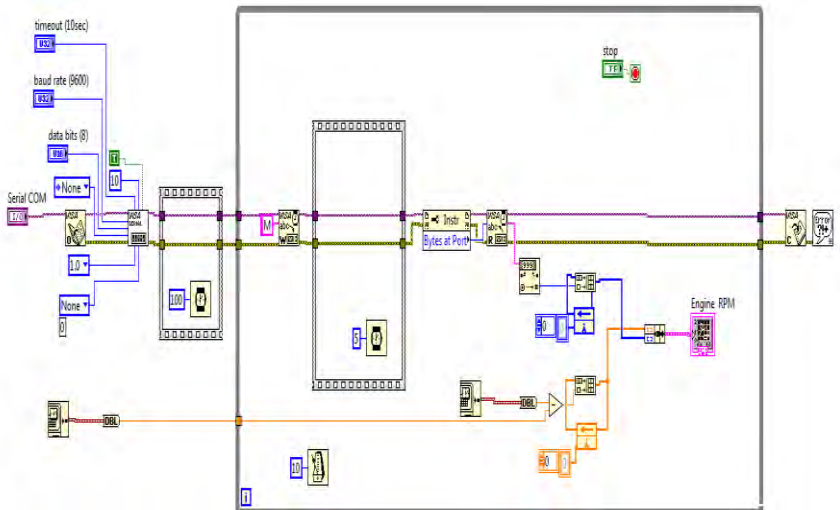
**Gambar 3.19** Tampilan Program Arduino

### 3.3.2 Software LabView

Perancangan dengan *software* LabView dilakukan untuk menampilkan data kecepatan motor DC dalam bentuk grafik. Selain itu



dengan LabView grafik data kecepatan juga dapat disimpan. Tampilan *software* LabView yang digunakan dapat dilihat pada **Gambar 3.20**.

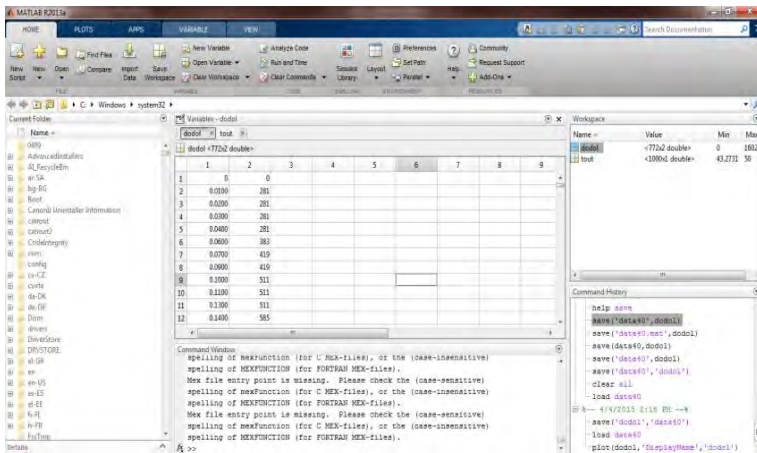


**Gambar 3.20** Tampilan *Software* LabView

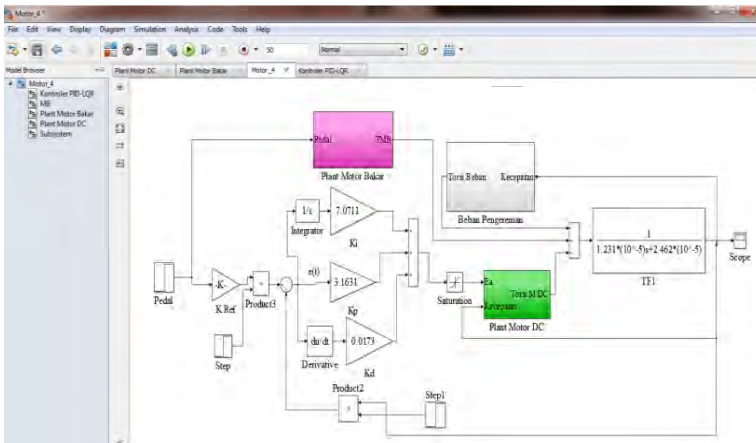
### 3.3.3 *Software* MATLAB

*Software* MATLAB yang digunakan pada Tugas Akhir ini adalah MATLAB R2013a. *Software* MATLAB digunakan untuk merancang simulasi sistem menggunakan menu Editor dan Simulink. MATLAB juga berfungsi sebagai *interface* antara mesin dan manusia atau *Human Machine Interface* (HMI) pada proses pengambilan data dan pengiriman data untuk ditampilkan dalam suatu grafik. Tampilan MATLAB R2013a dapat dilihat pada **Gambar 3.21** dan tampilan Simulink dari MATLAB dapat dilihat pada **Gambar 3.22**.

Pada Tugas Akhir ini, menu Editor digunakan untuk merancang program LQR untuk mendapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  dengan memasukkan matriks **A**, matriks **B**, dan matriks pembobot sedangkan menu Simulink digunakan untuk merancang diagram blok keseluruhan sistem, termasuk di dalamnya untuk merancang kontroler PID yang nilainya telah di-*tuning* melalui program LQR.



Gambar 3.21 Tampilan MATLAB R2013a



Gambar 3.22 Tampilan Simulink pada MATLAB R2013a

### 3.4 Pengujian Sistem

Untuk mengetahui performansi pada *plant*, karakteristik dan kelayakan dari sistem tersebut, maka dibutuhkan pengujian pada beberapa sistem pada simulator HEV.

### 3.4.1 Pengujian Sensor

#### 3.4.1.1 Pengujian Rotary Encoder

Pada Tugas Akhir ini untuk mengukur kecepatan putar motor DC digunakan sensor *rotary encoder*. Pengujian sensor *rotary encoder* dilakukan dengan membandingkan hasil nilai pembacaan kecepatan putar *rotary encoder* dengan nilai pembacaan kecepatan putar yang diukur menggunakan dengan *tachometer digital*. Tujuannya adalah untuk mengetahui keakurasian dari pembacaan kecepatan putar. Bentuk fisik *tachometer digital* dapat dilihat pada **Gambar 3.23**.



**Gambar 3.23** Bentuk Fisik *Tachometer Digital*

Berdasarkan hasil pengujian didapatkan hasil pembacaan keluaran *rotary encoder* dengan kecepatan putar pada poros motor DC menggunakan *tachometer digital*. Data hasil pembacaan terhadap sensor *rotary encoder* dan *tachometer digital* ditunjukkan pada **Tabel 3.3**. Dari hasil pengujian tersebut dapat diketahui perbandingan perubahan besar kecepatan putar motor DC dan kecepatan putar *rotary encoder* dengan mencari linearisasi menggunakan regresi linear.

Persamaan linearisasi antara kecepatan motor DC hasil pembacaan *rotary encoder* dengan kecepatan motor DC hasil pembacaan *tachometer digital* yang ditunjukkan pada **Persamaan (3.13)** akan dimasukkan pada program Arduino. Hal ini dilakukan agar data kecepatan yang diolah oleh LabView hanya memiliki sedikit perbedaan dengan data kecepatan asli. Perbandingan data kecepatan hasil linearisasi dengan kecepatan asli dapat dilihat pada **Tabel 3.4**.

**Tabel 3.3** Data Hasil Pembacaan terhadap Sensor *Rotary Encoder* dan *Tachometer Digital*

No	Vin (V)	Kecepatan <i>Tachometer Digital</i> (rpm)	Kecepatan <i>Rotary Encoder</i> (rpm)
1	25	595	570
2	50	981	1070
3	80	1136	1320
4	90	1179	1400
5	100	1226	1550
6	120	1339	1602
7	150	1539	1881
8	160	1606	1911
9	170	1676	2164
10	180	1743	2209
11	190	1806	2314

Hasil linierisasi menggunakan regresi linear dapat dilihat pada **Persamaan (3.13)**.

$$f(x) = 229,9536927 + 0,680455391x \quad (3.13)$$

dengan,

$f(x)$  : kecepatan *output* arduino (rpm)

$x$  : kecepatan *output rotary encoder* (rpm)

**Tabel 3.4** Perbandingan Data Kecepatan Hasil Linearisasi dengan Kecepatan Asli (dalam rpm)

Kecepatan untuk LabView (Hasil Linearisasi)	Kecepatan Asli ( <i>Tachometer Digital</i> )
592,3796066	595
981,3256628	981
1133,129503	1136

Kecepatan untuk Labview (Hasil Linearisasi)	Kecepatan Asli (Tachometer Digital)
1179,901497	1179
1229,955737	1226
1348,937125	1339
1531,922295	1539
1614,798986	1606
1677,982207	1676
1716,548588	1743
1819,118750	1806

### 3.4.1.2 Pengujian Sensor Potensiometer

Masukan dari sistem HEV direpresentasikan dalam bukaan pedal. Sensor potensiometer dipasang untuk mengetahui derajat bukaan *throttle* pada karburator. Dengan adanya potensiometer dapat direpresentasikan perubahan hambatan pada potensiometer berbanding lurus dengan perubahan presentase bukaan *throttle* ICE. Hubungan antar hambatan potensiometer dengan presentase bukaan *throttle* berdasarkan **Persamaan (3.14)**.

$$f(x) = \frac{100x - 1010}{-8,17} \quad (3.14)$$

dengan,

$f(x)$  : presentase bukaan *throttle* (%)  
 $x$  : resistansi potensiometer (Ohm)

## 3.5 Proses Identifikasi dan Pemodelan Sistem

Untuk melakukan pengaturan terhadap suatu sistem, hal pertama yang harus diketahui adalah karakteristik dan hubungan *input-output* dari sistem tersebut. Hal ini dapat dilakukan dengan melakukan identifikasi pada sistem. Identifikasi dan pemodelan yang dilakukan pada Tugas Akhir ini terbagi menjadi 2 bagian, yaitu motor DC, dan ICE. Identifikasi untuk motor DC dilakukan dengan menggunakan identifikasi fisik dari konfigurasi motor DC paralel. Sedangkan untuk

identifikasi ICE dilakukan dengan sinyal uji *step* menggunakan metode Strejc.

### 3.5.1 Identifikasi dan Pemodelan Motor DC

#### 3.5.1.1 Identifikasi Parameter Motor DC

Identifikasi parameter motor DC dilakukan untuk mendapatkan parameter-parameter dari sistem berdasarkan hasil pengukuran *input-output plant*. Parameter-parameter yang diperoleh akan digunakan untuk mendapatkan model matematika dari motor DC.

Untuk keperluan pemodelan motor DC dibutuhkan beberapa parameter, yaitu konstanta gaya gerak listrik balik motor DC ( $K_{ggl}$ ), konstanta torsi motor DC ( $K_{TMDC}$ ), induktansi kumparan jangkar ( $L_a$ ) dan resistansi kumparan jangkar ( $R_a$ ), induktansi kumparan medan ( $L_f$ ) dan resistansi kumparan medan ( $R_f$ ), dan parameter beban ( $J_m$ ,  $B_m$ , dan  $B_l$ ).

##### a. Parameter Konstanta GGL Balik Motor DC

Percobaan untuk mendapatkan parameter konstanta balik motor DC dilakukan dengan memutar motor DC dengan motor DC hanya diberi masukan pada kumparan medannya saja

Percobaan ini dilakukan pada 5 kondisi tegangan masukan ke kumparan medan, yaitu 150 V, 160 V, 170 V, 180 V dan 190 V. Perubahan tegangan masukan ini dilakukan pada 3 kondisi kecepatan yang berbeda, yaitu 1000 rpm, 910 rpm dan 830 rpm. Setelah semua data dari percobaan didapatkan, data dimasukkan kedalam **Persamaan (3.15)** untuk mendapatkan nilai konstanta balik motor DC ( $K_{ggl}$ ). Dari data hasil percobaan pada **Tabel 3.5** didapatkan nilai rata-rata konstanta balik motor DC ( $K_{ggl}$ ) sebesar 0,71546.

$$K_{ggl} = \frac{E_a}{I_f \times \omega} \quad (3.15)$$

dengan,

$K_{ggl}$  : Konstanta gaya gerak listrik balik motor DC (V-sec/rad)

$E_a$  : Tegangan kumparan jangkar (V)

$I_f$  : Arus medan (A)

$\omega$  : Kecepatan putar (rpm)

**Tabel 3.5** Data Percobaan Konstanta Balik Motor DC

No	$V_{in}$ (V)	Kecepatan (rpm)	$I_f$ (A)	$E_a$ (V)	$K_{ggl}$
1	150	1000	0,0729	68	0,9327
2	160	1000	0,0795	70	0,8805
3	170	1000	0,0835	72	0,8622
4	180	1000	0,0945	73	0,7724
5	190	1000	0,1006	75	0,7455
6	150	910	0,0778	54	0,6512
7	160	910	0,0816	59	0,6550
8	170	910	0,0866	59	0,6812
9	180	910	0,0916	60	0,7230
10	190	910	0,0952	62	0,6940
11	150	830	0,0767	49	0,6388
12	160	830	0,0822	52	0,6236
13	170	830	0,0864	56	0,6481
14	180	830	0,0924	57	0,6168
15	190	830	0,0972	59	0,6069
Rata-rata $K_{ggl}$					<b>0.71546</b>

**b. Parameter Konstanta Torsi Motor DC**

Untuk mendapatkan parameter konstanta motor DC dibutuhkan beberapa parameter yaitu arus jangkar ( $I_a$ ), arus medan ( $I_f$ ), dan perubahan torsi motor DC ( $\Delta\tau_{mdc}$ ). Konstanta torsi motor DC ( $K_{TMDC}$ ) diperoleh berdasarkan **Persamaan (3.16)**.

$$K_{TMDC} = \frac{\Delta\tau_{mdc}}{(I_{a\ max} \times I_{f\ max}) - (I_{a\ nom} \times I_{f\ nom})} \quad (3.16)$$

dengan,

- $K_{TMDC}$  : Konstanta torsi motor DC (V-sec/rad)  
 $\Delta\tau_{mdc}$  : Perubahan torsi motor DC (N-m)  
 $I_{a\ max}$  : Arus kumparan jangkar kondisi maksimal (A)  
 $I_{f\ max}$  : Arus medan kondisi maksimal (A)  
 $I_{a\ nom}$  : Arus kumparan jangkar kondisi nominal (A)  
 $I_{f\ nom}$  : Arus medan kondisi nominal (A)

Percobaan untuk mendapatkan parameter konstanta motor DC dilakukan pada kondisi maksimal dan nominal. Pada percobaan kondisi maksimal, motor DC dihidupkan dengan sumber tegangan sebesar 190 V dan beban rem elektromagnetik sebesar 80 V. Percobaan kondisi nominal dilakukan dengan cara menghidupkan motor DC dengan sumber tegangan sebesar 150 V dan beban rem elektromagnetik sebesar 40 V. Dari hasil percobaan didapatkan nilai arus jangkar ( $I_a$ ), dan arus medan ( $I_f$ ) pada saat kondisi maksimal dan kondisi nominal. Data hasil percobaan dapat dilihat pada **Tabel 3.6**.

**Tabel 3.6** Data Percobaan Kondisi Maksimal dan Nominal

Parameter	Nilai (A)
$I_{a\ max}$	6,4200
$I_{a\ nom}$	4,2800
$I_{f\ max}$	0,0763
$I_{f\ nom}$	0,0631

Setelah didapatkan nilai dari parameter arus jangkar ( $I_a$ ), dan arus medan ( $I_f$ ), langkah selanjutnya adalah mendapatkan nilai dari perubahan torsi motor ( $\Delta\tau_{mdc}$ ).  $\Delta\tau_{mdc}$  dapat diperoleh berdasarkan **Persamaan (3.17)** sebagai berikut :

$$P_{in} = V_{in}(I_a + I_f) = P_{rugi-rugi} + \tau_m \times \omega \quad (3.17)$$

dengan,

- $P_{in}$  : Daya *input* motor DC (Watt)  
 $V_{in}$  : Tegangan *input* motor DC (V)



- $I_a$  : Arus kumparan jangkar (A)  
 $I_f$  : Arus medan (A)  
 $P_{rugi-rugi}$  : Daya rugi-rugi motor DC (Watt)  
 $\tau_m$  : Torsi motor DC (N-m)  
 $\omega$  : Kecepatan putar motor DC (rpm)

Percobaan pada kondisi maksimal dan minimal dilakukan hingga motor DC memiliki kecepatan 1450 rpm. Karena percobaan dilakukan pada kondisi kecepatan yang sama, maka nilai  $P_{rugi-rugi}$  pada kondisi maksimal dan minimal juga sama, sehingga dapat dieliminasi. **Persamaan (3.17)** dapat diubah menjadi **Persamaan (3.18)**.

$$\omega(\tau_{m \max} - \tau_{m \text{ nom}}) = V_{in \max} (I_{a \max} + I_{f \max}) - V_{in \text{ nom}} (I_{a \text{ nom}} + I_{f \text{ nom}}) \quad (3.18)$$

$$\tau_{mdc} = \frac{V_{in \max} (I_{a \max} + I_{f \max}) - V_{in \text{ nom}} (I_{a \max} + I_{f \max})}{\omega} \quad (3.19)$$

dengan,

- $V_{in \max}$  : tegangan *input* motor DC saat beban 80 V (V)  
 $I_{a \max}$  : arus kumparan jangkar saat beban 80 V (A)  
 $I_{f \max}$  : arus medan saat beban 80 V (A)  
 $V_{in \text{ nom}}$  : tegangan *input* motor DC saat beban 40 V (V)  
 $I_{a \text{ nom}}$  : arus kumparan jangkar saat beban 40 V (A)  
 $I_{f \text{ nom}}$  : arus medan saat beban 40 V (V)

Data hasil pengukuran parameter arus jangkar, arus medan, dan tegangan input saat kondisi maksimal dan nominal dapat dilihat pada **Tabel 3.7**. Substitusi data hasil pengukuran pada **Tabel 3.7** ke dalam Persamaan (3.19).

$$\Delta\tau_{mdc} = \frac{V_{in \max} (I_{a \max} + I_{f \max}) - V_{in \text{ nom}} (I_{a \text{ nom}} + I_{f \text{ nom}})}{\omega}$$

$$\Delta\tau_{mdc} = \frac{63,5(6,42+0,0763)-49(4,27+0,0631)}{1450}$$

$$\Delta\tau_{mdc} = 0,138$$

**Tabel 3.7** Data Hasil Pengukuran Parameter

Parameter	Nilai
$I_a \max$	6,4200 A
$I_a \text{ nom}$	4,2700A
$I_f \max$	0,0763 A
$I_f \text{ nom}$	0,0631 A
$V_{in \max}$	63,500 V
$V_{in \text{ nom}}$	49,000 V

Setelah mendapatkan nilai  $\Delta\tau_{mdc}$ , substitusikan nilai  $\Delta\tau_{mdc}$  dan nilai pada **Tabel 3.7** pada **Persamaan (3.16)** sehingga didapatkan nilai dari konstanta torsi motor DC ( $K_{TMDC}$ ).

$$K_{TMDC} = \frac{\Delta\tau_{mdc}}{(I_a \max \times I_f \max) - (I_a \text{ nom} \times I_f \text{ nom})}$$

$$K_{TMDC} = \frac{0,138}{(6,42 \times 0,0763) - (4,27 \times 0,063)}$$

$$K_{TMDC} = \mathbf{0,624}$$

### c. Induktansi dan Resistansi

Untuk keperluan pemodelan motor DC, parameter yang dibutuhkan selanjutnya adalah nilai induktansi dan resistansi dari kumparan motor, baik kumparan jangkar maupun medan ( $L_a, R_a, L_f, \& R_f$ ). Pengukuran parameter ini dilakukan dengan menggunakan alat LCR meter. Hasil pengukuran parameter  $L_a, R_a, L_f, R_f$  dapat dilihat pada **Tabel 3.8**.

**Tabel 3.8** Hasil pengukuran parameter  $L_a, R_a, L_f, R_f$

Parameter	Nilai
$R_a$	19,748 $\Omega$
$R_f$	2377,120 $\Omega$
$L_a$	0,091160 H
$L_f$	23,34 H

#### d. Parameter Beban

Beban pada motor DC pada terbagi menjadi beban rem elektromagnetik dan beban komponen penghubung antara ICE dengan motor DC. Parameter-parameter yang dibutuhkan yaitu konstanta rem elektromagnetik ( $K_d$ ), momen inersia beban ( $J_m$ ), koefisien gesekan viskos beban ( $B_m$ ), dan beban rem elektromagnetik ( $B_l$ ).

Parameter konstanta rem elektromagnetik ( $K_d$ ), koefisien gesekan viskos beban ( $B_m$ ) didapatkan berdasarkan **Persamaan (3.20)**.

$$\omega \times B_m - \omega \times I_l \times K_d = K_{TMDC} \times I_a \times I_f \quad (3.20)$$

dengan,

- $\omega$  : Kecepatan putar motor DC (rpm)
- $B_m$  : Koefisien gesek viskos beban (N-m/rad/s)
- $I_l$  : Arus rem elektromagnetik (A)
- $K_d$  : Konstanta rem elektromagnetik
- $K_{TMDC}$  : Konstanta torsi motor DC (N-m/A)
- $I_a$  : Arus kumpar jangkar (A)
- $I_f$  : Arus medan (A)

Percobaan untuk mendapatkan parameter beban juga dilakukan pada kondisi maksimal (beban 80 V) dan kondisi nominal (beban 40 V) , namun untuk mendapatkan parameter beban harus diketahui arus rem elektromagnetik. Berikut ini merupakan data hasil pengukuran arus rem elektromagnetik pada **Tabel 3.9**.

**Tabel 3.9** Data Hasil Pengukuran Arus Rem Elektromagnetik

Beban (V)	$I_l$ (A)	$\Omega$ (rpm)
40	0,56	1450
80	1,14	1450

Parameter pada **Tabel 3.9** disubstitusikan ke dalam **Persamaan (3.20)**, sehingga didapatkan **Persamaan (3.21)** dan **Persamaan (3.22)**.

saat beban 80 V ( $I_l = 1,14$ ), sehingga didapatkan :

$$\begin{aligned} 1450B_m - 1450 \times 1,14 \times K_d &= 0,624 \times 6,42 \times 0,0763 \\ 1450B_m - 1653K_d &= 0,3057 \end{aligned} \quad (3.21)$$

saat beban 40 V ( $I_l = 0,56$ ), sehingga didapatkan :

$$\begin{aligned} 1450B_m - 1450 \times 0,56 \times K_d &= 0,624 \times 4,27 \times 0,0631 \\ 1450B_m - 812K_d &= 0,1681 \end{aligned} \quad (3.22)$$

eliminasi **Persamaan (3.21)** dan **Persamaan (3.22)**, sehingga didapatkan nilai  $K_d = -1,636 \times 10^{-4}$  dan  $B_m = 2,462 \times 10^{-5}$ .

Koefisien gesek viskom rem elektromagnetik diperoleh berdasarkan **Persamaan (3.23)** sebagai berikut :

$$B_l = -K_d \times I_l \quad (3.23)$$

Beban yang digunakan pada Tugas Akhir ini yaitu pada saat kondisi nominal, dimana arus rem elektromagnetik ( $I_l$ ) sebesar 0,56 A. Dengan mensubstitusikan arus rem dan nilai  $K_d$  yang telah didapatkan ke **Persamaan (3.23)**, diperoleh :

$$\begin{aligned} B_l &= -(-1,636 \times 10^{-4}) \times 0,56 \\ B_l &= -(-1,636 \times 10^{-4}) \times 0,56 \\ B_l &= 0,000091616 \text{ N-m/rad/sec} \end{aligned}$$

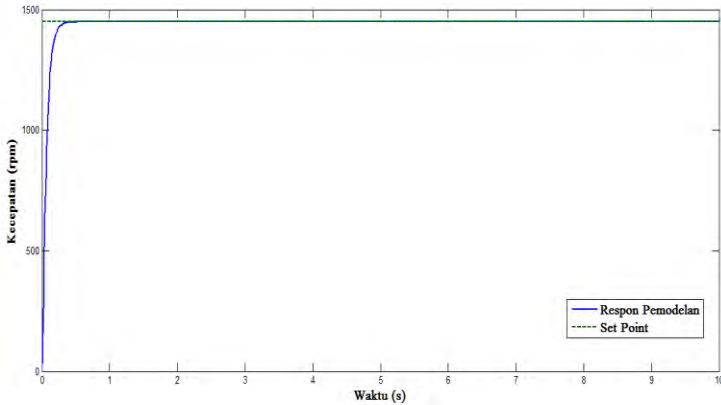
Parameter momen inersia beban ( $J_m$ ) didapatkan dari respon sistem motor DC dengan mengambil data respon kecepatan dari motor DC saat diberikan *input* tegangan berbentuk sinyal *step* seperti pada **Gambar 3.24**. Sinyal *step* disini adalah tegangan dengan perubahan dari 0 V sampai 150 V dalam waktu yang singkat. *Transfer function* beban dapat dilihat berdasarkan **Persamaan (3.24)** :

$$\frac{1}{J_m s + B_m} = \frac{1/B_m}{J_m/B_m s + 1} = \frac{1}{\tau s + 1} \quad (3.24)$$

sehingga,

$$\tau = \frac{J_m}{B_m} \quad (3.25)$$

$$J_m = \tau \times B_m \quad (3.26)$$



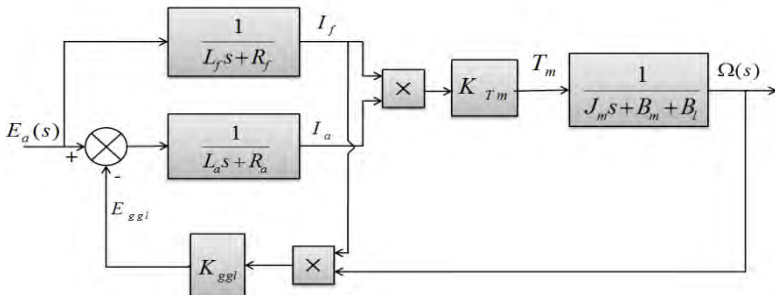
**Gambar 3.24** Respon Motor DC saat Diberikan Input Sinyal *Step*

Berdasarkan respon pada **Gambar 3.24** didapatkan nilai  $\tau = 0,5$ . Diketahui nilai  $B_m = 2,462 \times 10^{-5}$  N-m/rad/s, sehingga nilai  $J_m$  dapat diperoleh dengan mensubstitusi kedua nilai tersebut pada **Persamaan (3.26)**.

$$J_m = 0,5 \times 2,462 \times 10^{-5}$$

$$J_m = 1,231 \times 10^{-5} \text{ kg-m}^2$$

. Parameter motor DC yang telah didapatkan dimasukkan ke dalam diagram blok motor DC seperti yang ditunjukkan yang ditunjukkan **Gambar 3.25**.

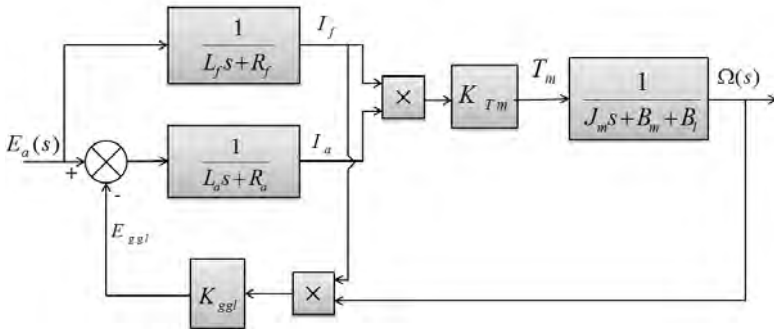


**Gambar 3.25** Diagram Blok Motor DC

Variasi beban yang diperbolehkan dalam pengujian kontroler harus sesuai dengan pemodelan nominal untuk ICE (d disesuaikan dengan *range* pembebanan nominal), sehingga dibatasi beban yang diperbolehkan adalah pada *range* arus beban 0,56 A– 1,14 A.

### 3.5.1.2 Pemodelan Motor DC

Pada Tugas Akhir ini digunakan Motor DC dengan Konfigurasi Paralel. Motor DC dengan konfigurasi paralel memiliki karakteristik non linear terhadap arus jangkar, arus medan, dan torsi motor. Diagram blok fisik motor DC konfigurasi paralel dapat dilihat pada **Gambar 3.26**.



**Gambar 3.26** Diagram Blok Fisik Motor DC Konfigurasi Paralel

Karena terdapat beberapa parameter pada konfigurasi motor DC paralel memiliki karakteristik non linier, untuk keperluan perancangan kontroler yang memerlukan model linear maka beberapa persamaan pada motor DC harus dilinearisasi yaitu torsi motor ( $T_m$ ) dan konstanta balik motor ( $E_{ggl}$ ).

- a. Linearisasi Persamaan Torsi Motor

$$T_m = K_{TM} I_a I_f \quad (3.27)$$

$$\tilde{T}_m = \left. \frac{\partial (K_{TM} I_a I_f)}{\partial I_a} \right|_{i_f = I_{fop}} \tilde{i}_a + \left. \frac{\partial (K_{TM} I_a I_f)}{\partial I_f} \right|_{i_a = I_{aop}} \tilde{i}_f$$

$$\tilde{T}_m = K_{TM} I_{fop} \tilde{I}_a + K_{TM} I_{aop} \tilde{I}_f \quad (3.28)$$

- b. Linearisasi Persamaan  $E_{ggl}$

$$E_{ggl} = K_{ggl} I_f \Omega \quad (3.29)$$

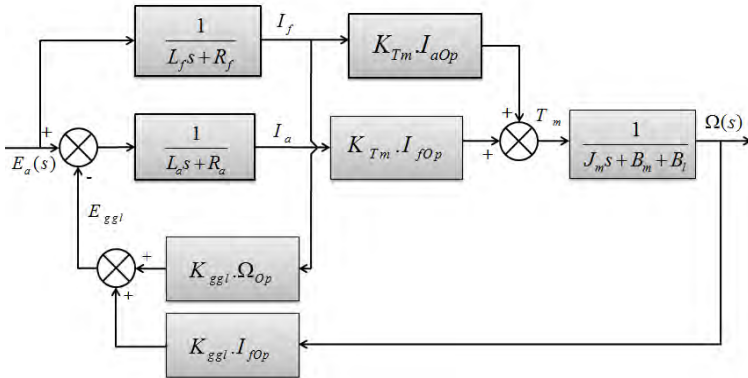
$$\check{E}_{ggl} = \frac{\partial(K_{ggl} I_f \Omega)}{\partial I_f} \Big|_{\Omega=\Omega_{op}} \check{i}_f + \frac{\partial(K_{ggl} I_f \Omega)}{\partial \Omega} \Big|_{i_f=I_{fop}} \check{\Omega}$$

$$\check{E}_{ggl} = K_{ggl} I_f \Omega_{op} + K_{ggl} I_{fop} \Omega \quad (3.30)$$

dengan,

- $T_m$  : Torsi motor DC (N-m)
- $K_{TM}$  : Konstanta torsi motor (N-m/A)
- $L_a$  : Induktansi kumparan jangkar (H)
- $R_a$  : Reaktansi kumparan jangkar ( $\Omega$ )
- $L_f$  : Induktansi medan (H)
- $R_f$  : Reaktansi medan ( $\Omega$ )
- $E_{ggl}$  : Gaya gerak listrik balik (V)
- $K_{ggl}$  : Konstanta gaya gerak listrik balik (V-sec/rad)
- $I_{aop}$  : Arus kumparan jangkar beban nominal (A)
- $I_{fop}$  : Arus medan saat beban nominal (A)
- $\Omega_{op}$  : Kecepatan putar saat beban nominal (rpm)

Setelah beberapa persamaan yang non linear dilinearisasi, maka **Gambar 3.26** dapat diubah menjadi **Gambar 3.27**.



**Gambar 3.27** Diagram Blok Fisik Motor DC Konfigurasi Paralel Hasil Linearisasi

Dari **Gambar 3.27** dapat dicari model matematika motor DC melalui beberapa persamaan :

$$T_m = \frac{K_{TM} I_{fop}}{L_a s + R_a} (E_a - E_{ggl}) + \frac{K_{TM} I_{aop}}{L_f s + R_f} E_a \quad (3.31)$$

$$E_{ggl} = \frac{K_{ggl} \Omega_{op}}{L_f s + R_f} E_a + (K_{ggl} I_{fop} \Omega) \quad (3.32)$$

substitusi **Persamaan (3.32)** ke **Persaman (3.31)**, sehingga didapat **Persamaan (3.33)** sebagai berikut :

$$\begin{aligned} T_m &= \frac{K_{TM} I_{fop}}{L_a s + R_a} \left( E_a - \left( \frac{K_{ggl} \Omega_{op}}{L_f s + R_f} E_a + K_{ggl} I_{fop} \Omega \right) \right) + \frac{K_{TM} I_{aop}}{L_f s + R_f} E_a \\ T_m &= \frac{K_{TM} I_{aop}}{L_f s + R_f} E_a + \frac{K_{TM} I_{fop}}{L_a s + R_a} \left( E_a - \frac{K_{ggl} \Omega_{op}}{L_f s + R_f} E_a - K_{ggl} I_{fop} \Omega \right) \\ T_m &= E_a \left[ \frac{K_{TM} I_{aop}}{L_f s + R_f} + \frac{K_{TM} I_{fop}}{L_a s + R_a} - \frac{K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s + R_f)(L_a s + R_a)} \right] - \underbrace{\frac{K_{ggl} I_{fop} \Omega K_{TM} I_{fop}}{L_a s + R_a}}_{disturbance} \\ \frac{T_m}{E_a} &= \frac{K_{TM} I_{aop}}{L_f s + R_f} + \frac{K_{TM} I_{fop}}{L_a s + R_a} - \frac{K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s + R_f)(L_a s + R_a)} \\ \frac{T_m}{E_a} &= \frac{K_{TM} I_{aop} (L_a s + R_a) + K_{TM} I_{fop} (L_f s + R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s + R_f)(L_a s + R_a)} \quad (3.33) \end{aligned}$$

Masukkan nilai parameter motor DC yang telah didapatkan pada perhitungan sebelumnya ke dalam **Persamaan (3.33)**, sehingga didapatkan hubungan *input-output* dalam fungsi alih pada **Persamaan (3.34)** sebagai berikut :

$$\frac{T_m}{E_a} = \frac{50(0,01s+1)}{s^2+321,15s+22249,9} \quad (3.34)$$

Nilai  $\Omega$  dapat diperoleh dengan menggunakan **Persamaan (3.35)** :

$$\Omega = T_m \times \frac{1}{J_m s + B_m + B_l} \quad (3.35)$$



$$\Omega = \left[ \frac{K_{TM} I_{aop} (L_a s + R_a) + K_{TM} I_{fop} (L_f s + R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s + R_f)(L_a s + R_a)} \right] E_a \cdot \frac{1}{J_m s + B_m + B_l}$$

$$\frac{\Omega}{E_a} = \frac{K_{TM} I_{aop} (L_a s + R_a) + K_{TM} I_{fop} (L_f s + R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s + R_f)(L_a s + R_a)(J_m s + B_T)} \quad (3.36)$$

dengan,

- $T_m$  : Torsi motor (N-m)
- $L_a$  : Induktansi kumparan jangkar (H)
- $R_a$  : Reaktansi kumparan jangkar ( $\Omega$ )
- $L_f$  : Induktansi medan (H)
- $R_f$  : Reaktansi medan ( $\Omega$ )
- $K_{ggl}$  : Konstanta gaya gerak listrik balik (V-s/rad)
- $I_{aop}$  : Arus kumparan jangkar saat beban nominal (A)
- $I_{fop}$  : Arus medan saat beban nominal (A)
- $\Omega_{op}$  : Kecepatan putar saat beban nominal (rpm)

Masukkan nilai parameter yang telah didapatkan pada perhitungan sebelumnya ke dalam **Persamaan (3.36)**, sehingga didapatkan hubungan *input-output* dalam fungsi alih pada **Persamaan (3.37)** sebagai berikut :

$$\frac{\Omega}{E_a} = \frac{1,16s + 105,491}{0,000026191 s^3 + 0,00858 s^2 + 0,65663 s + 5,45650906} \quad (3.37)$$

karena nilai  $L_f \cdot L_a \cdot J_m = 0,000026191$  sangat kecil maka dapat diabaikan, sehingga **Persamaan (3.37)** menjadi sebagai berikut:

$$\frac{\Omega}{E_a} = \frac{1,16s + 105,491}{0,00858 s^2 + 0,65663 s + 5,45650906} \quad (3.38)$$

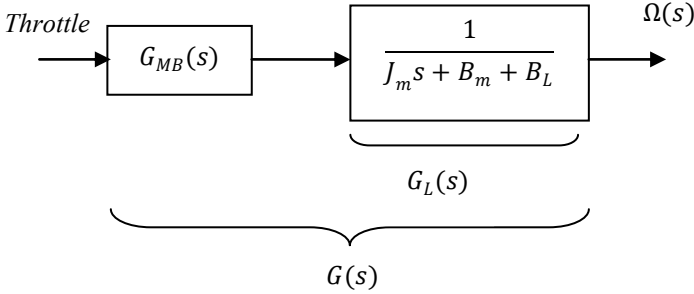
$$\frac{\Omega}{E_a} = \frac{12295,58(0,01s + 1)}{s^2 + 76,4785 s + 635,9768} \quad (3.39)$$

Pada **Persamaan (3.39)** diatas terdapat *time delay* 0,01 detik. Karena nilainya kecil maka dapat diabaikan, sehingga **Persamaan (3.39)** dapat diubah menjadi sebagai berikut :

$$G(s) = \frac{\Omega}{E_a} = \frac{12295,58}{s^2 + 76,4785s + 635,9768} \quad (3.40)$$

### 3.5.2 Identifikasi dan Pemodelan ICE

Untuk mendapatkan permodelan ICE, dilakukan identifikasi statis menggunakan metode Strejc. Proses pemodelan Strejc memanfaatkan grafik respon dari ICE saat diberikan masukan berupa sinyal *step*. Dari grafik respon akan didapatkan parameter yang akan menghasilkan model matematika yang mendekati respon ICE sebenarnya. Hasil pemodelan dengan metode Strejc adalah pendekatan dari pemodelan ICE yang telah terhubung dengan beban seperti yang dijelaskan pada **Gambar 3.28**.



**Gambar 3.28** Diagram Blok Pemodelan Strejc

sehingga didapatkan **Persamaan (3.41)**,

$$G_{MB}(s) = \frac{G(s)}{G_L(s)} \quad (3.41)$$

Pemodelan ICE terbagi menjadi 3, yaitu saat pembebanan minimal, nominal dan pembebanan maksimal.

#### a. Pembebanan Minimal

Pemodelan dengan pembebanan minimal dilakukan tanpa diberikan tambahan beban dari rem elektromagnetik. Tetapi untuk pengukuran kecepatan putar ICE membutuhkan penyatuan poros dengan motor DC yang telah ter-*couple* dengan *rotary encoder* sebagai pembacaan sensor, sehingga poros motor DC yang telah dihubungkan dengan ICE menggunakan gir dapat dianggap sebagai beban minimal

untuk ICE berbeban. Nilai  $X_{ss}$  merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan untuk ICE beban nominal dapat dilihat pada **Gambar 3.29**. Dari gambar tersebut kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$\begin{aligned} Y_{ss} &= 1523 \\ X_{ss} &= 41,2 \\ K &= \frac{Y_{ss}}{X_{ss}} = \frac{1523}{41,2} = 36,97 \end{aligned}$$

Dari grafik respon pada **Gambar 3.29** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat kan nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$\begin{aligned} T_U &= 0,01 \\ T_U + T_N &= 0,41 \\ T_N &= 0,4 \end{aligned}$$

setelah itu dapat dicari nilai para meter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\begin{aligned} \tau &= \frac{T_U}{T_N} \\ \tau &= 0,025 \end{aligned}$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i = 0,264$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$\begin{aligned} y_i &= 0,264 \times y_{ss} \\ y_i &= 402 \\ t_i &= 0,135 \end{aligned}$$

Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.

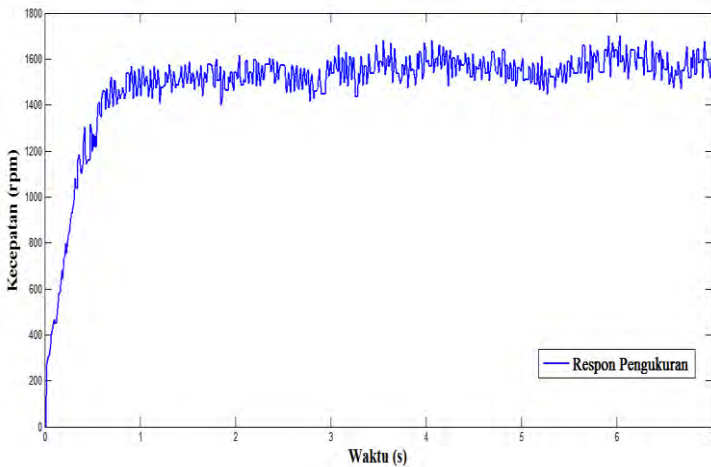
$$\tau_{ST} = \frac{t_i}{n-1} = \frac{0,135}{2-1}$$

$$\tau_{ST} = 0,135$$

sehingga didapat pendekatan model orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST}(s) = \frac{K}{(\tau_{ST}s+1)^2}$$

$$G(s) = \frac{36,97}{(0,135s+1)^2}$$



**Gambar 3.29** Respon Kecepatan Putar ICE Beban Minimal

#### b. Pembebanan Nominal

Pembebanan nominal dilakukan dengan memberikan tambahan beban dari rem elektromagnetik. Setelah dilakukan beberapa kali percobaan, maka ditentukan bahwa beban nominal adalah beban dengan tegangan input pada rem elektromagnetik sebesar 40 V. Nilai  $X_{ss}$  merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan untuk ICE beban nominal dapat dilihat pada **Gambar 3.30**. Dari gambar tersebut

kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$\begin{aligned} Y_{ss} &= 1448 \\ X_{ss} &= 41,2 \\ K &= \frac{Y_{ss}}{X_{ss}} = \frac{1450}{41,25} = 35,14 \end{aligned}$$

Dari grafik respon pada **Gambar 3.30** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat kan nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$\begin{aligned} T_U &= 0,007 \\ T_U + T_N &= 0,407 \\ T_N &= 0,4 \end{aligned}$$

setelah itu dapat dicari nilai para meter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\begin{aligned} \tau &= \frac{T_U}{T_N} \\ \tau &= 0,0175 \end{aligned}$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i = 0,264$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$\begin{aligned} y_i &= 0.264 \times y_{ss} \\ y_i &= 376 \\ t_i &= 0,125 \end{aligned}$$

Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.

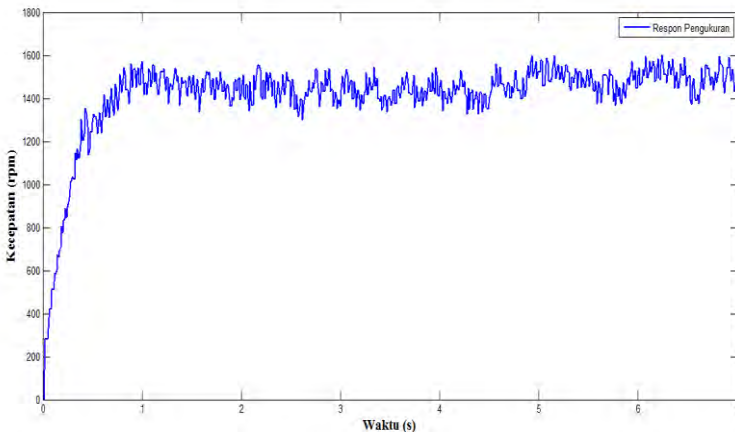
$$\tau_{ST} = \frac{t_i}{n-1} = \frac{0,125}{2-1}$$

$$\tau_{ST} = 0,125$$

sehingga didapat pendekatan mode orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST}(s) = \frac{K}{(\tau_{ST}s+1)^2}$$

$$G(s) = \frac{35,14}{(0,125s+1)^2}$$



**Gambar 3.30 Respon Kecepatan Putar ICE Beban Nominal**

### c. Metode Pembebanan Maksimal

Pembebanan maksimal dilakukan dengan memberi beban pada ICE berupa rem elektromagnetik. Setelah dilakukan beberapa percobaan, maka ditentukan bahwa beban nominal adalah beban dengan tegangan input pada rem elektromagnetik sebesar 80 V. Nilai  $X_{ss}$  merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan ICE beban maksimal dapat dilihat pada **Gambar 3.31**. Dari gambar tersebut kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$Y_{ss} = 1323$$

$$X_{ss} = 41,2$$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1323}{41.25} = 32,11$$

Dari grafik respon pada **Gambar 3.31** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$T_U = 0,01$$

$$T_U + T_N = 0,45$$

$$T_N = 0,14$$

setelah itu dapat dicari nilai para meter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\tau = \frac{T_u}{T_N} = 0,07$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i = 0,264$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$y_i = 0,264 \times y_{ss}$$

$$y_i = 350$$

$$t_i = 0,130$$

Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.

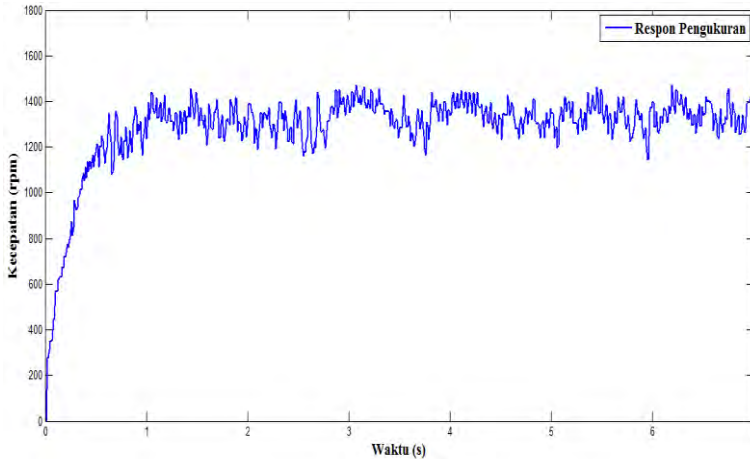
$$\tau_{ST} = \frac{t_i}{n-1} = \frac{0,130}{2-1}$$

$$\tau_{ST} = 0,130$$

sehingga didapat pendekatan mode orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST(s)} = \frac{K}{(\tau_{ST}s+1)^2}$$

$$G(s) = \frac{32,11}{(0,130s+1)^2}$$



**Gambar 3.31** Respon Kecepatan Putar ICE Beban Maksimal

### 3.6 Pengujian dan Validasi

Setelah didapatkan beberapa model matematika pada *plant*, kemudian dilakukan pengujian dan validasi model matematika dengan cara membandingkan antara hasil perhitungan dengan hasil pengukuran respon. Hasil perbandingan tersebut akan divalidasi dengan menggunakan RMSE. Kemudian dihitung nilai *Relatif* RMSE (RRMSE) terhadap  $Y_{ss}$  dalam presentase. Pada kondisi sebenarnya, poros dari kedua penggerak ICE dan motor DC saling terhubung dari awal mula kerja sistem. Pada Tugas Akhir ini, karena pemodelan motor DC berbeban nominal sehingga pengujian dan validasi model serta pemodelan *plant* ICE yang dipilih adalah ICE berbeban nominal. Proses perhitungan identifikasi statis untuk mendapatkan pemodelan terhadap *plant* dilakukan sebanyak 5 kali kemudian dilakukan validasi pada model yang didapatkan.

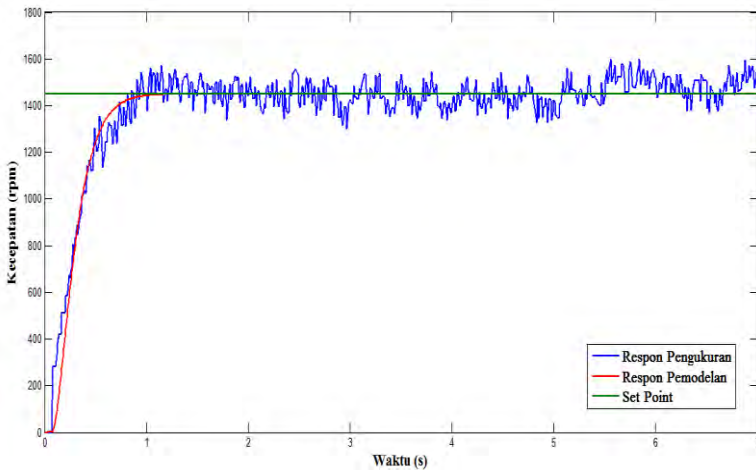
Persamaan model dan validasi *plant* ICE beban nominal dapat dilihat pada **Tabel 3.10**, sedangkan **Gambar 3.32** merupakan hasil Identifikasi dan pemodelan ICE beban nominal.



**Tabel 3.10** Persamaan Model dan Validasi *Plant* ICE Beban Nominal

No	Persamaan Model <i>Plant</i>	RRMSE
1	$G(s) = \frac{35,00}{(0,128s+1)^2}$	10,08 %
2	$G(s) = \frac{35,20}{(0,128s+1)^2}$	10,04 %
3	$G(s) = \frac{35,14}{(0,125s+1)^2}$	9,90 %
4	$G(s) = \frac{34,98}{(0,129s+1)^2}$	10,14 %
5	$G(s) = \frac{34,80}{(0,124s+1)^2}$	9,94 %

Hasil respon *plant* motor DC yang dipilih dengan menggunakan metode Strejc kemudian dibandingkan dengan metode identifikasi lainnya. Setelah melakukan beberapa pengujian dengan beberapa metode dapat dilihat bahwa RMSE terkecil dapat dicapai dengan menggunakan metode Strejc sehingga dapat disimpulkan bahwa hasil *transfer function* motor DC dengan menggunakan metode Strejc lebih mirip dengan respon asli. Hasil pengujian metode pendekatan model dapat dilihat pada **Tabel 3.11**.



**Gambar 3.32** Hasil Identifikasi dan Pemodelan ICE Beban Nominal

**Tabel 3.11** Pengujian Metode Pendekatan Model

No	Metode	Persamaan Model <i>Plant</i>	RRMSE
1	Latzel	$G(s) = \frac{35,14}{(0,2151s+1)^2}$	15,81 %
2	Viteckova 2 <sup>nd</sup> Order	$G(s) = \frac{35,14}{(0,158s+1)^2}$	11,94 %
3	Strejc	$G(s) = \frac{35,14}{(0,125s+1)^2}$	9,90 %

Sehingga didapatkan model matematika ICE berdasarkan **Persamaan (3.41)** :

$$G_{MB}(s) = \frac{G(s)}{G_L(s)}$$

$$G_{MB}(s) = \frac{\frac{35,14}{(0,125s+1)^2}}{\frac{1}{(1,231 \times 10^{-5})s + (2,462 \times 10^{-5}) + (9,1616 \times 10^{-5})}}$$

$$G_{MB}(s) = \frac{(43,25734 \times 10^{-5})s + 408,453304 \times 10^{-5}}{0,015625s^2 + 0,25s + 1}$$

### 3.7 Perancangan Kontroler PID-LQR.

Model matematika motor DC dari proses identifikasi seperti ditunjukkan pada **Persamaan (3.40)**. Diketahui bahwa *plant* merupakan orde dua dengan input  $u(t)$ , output  $y(t)$ , dan fungsi alih *plant* sebagai berikut :

$$\frac{Y(s)}{U(s)} = \frac{a}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{-E(s)}{U(s)} \quad (3.42)$$

$$\frac{Y(s)}{U(s)} = \frac{12295,58}{s^2 + 76,4785s + 635,9768} = \frac{-E(s)}{U(s)} \quad (3.43)$$

$$E(s)[s^2 + 76,4785s + 635,9768] = -12295,58U(s)$$

$$s^2E(s) + 76,4785sE(s) + 635,9768E(s) = -12295,58U(s)$$

$$\ddot{e} + 76,4785\dot{e} + 635,9768e = -12295,58u$$

$$\ddot{e} = -76,4785\dot{e} - 635,9768e - 12295,58u \quad (3.44)$$

dimisalkan sebuah *state* baru,

$$x_0 = \int e(t)dt, \quad x_1 = e(t), \quad x_2 = \frac{de(t)}{dt}$$

$$\dot{x}_0 = e = x_1 \quad (3.45)$$

$$\dot{x}_1 = \dot{e} = x_2 \quad (3.46)$$

$$\dot{x}_2 = \ddot{e} \quad (3.47)$$

dengan,

$x_0 = \text{Augmented State}$

$x_1 = \text{Kecepatan putar motor DC}$

$x_2 = \text{Percepatan motor DC}$

Masukkan **Persamaan (3.44)** ke **Persamaan (3.47)**, sehingga didapatkan **Persamaan (3.48)** sebagai berikut :

$$\begin{aligned} \dot{x}_2 &= \ddot{e} \\ \dot{x}_2 &= -76,4785\dot{e} - 635,9768e - 12295,58u \\ \dot{x}_2 &= -76,4785x_2 - 635,9768x_1 - 12295,58u \\ \dot{x}_2 &= -635,9768x_1 - 76,4785x_2 - 12295,58u \end{aligned} \quad (3.48)$$

representasi *plant* dalam *state space* dapat dibentuk sebagai berikut:

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -635,9768 & -76,4785 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -12295,58 \end{bmatrix} u$$

sehingga diperoleh parameter *plant* sebagai berikut:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -635,9768 & -76,4785 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -12295,58 \end{bmatrix};$$

Setelah diperoleh parameter *plant*, ditentukan matriks pembobot **Q** dan **R** melalui *trial and error*. Matriks **A**, matriks **B**, matriks pembobot

**Q** dan **R** digunakan untuk memperoleh matriks Riccati **P** dengan menyelesaikan persamaan *Algebraic Riccati Equation* (ARE) seperti pada **Persamaan (2.47)**. Pada Tugas Akhir ini dilakukan pemilihan matriks **Q** dan **R** sebanyak 4 kali dengan matriks **Q** semidefinit positif dan matriks **R** definit positif sehingga menghasilkan 4 nilai matriks Riccati **P** yang berbeda-beda. Nilai matriks Riccati **P** melalui penyelesaian ARE ditunjukkan pada **Tabel 3.12**.

**Tabel 3.12** Nilai Matriks **P** Melalui Penyelesaian ARE

Matriks Pembobot		Matriks P
R	Q	
[0,1]	$\begin{bmatrix} 0,005 & 0 & 0 \\ 0 & 0,001 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0,0028 & 0,0002 & 0,0000 \\ 0,0002 & 0,0001 & 0,0000 \\ 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$
[0,1]	$\begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0,0104 & 0,0003 & 0,0000 \\ 0,0003 & 0,0003 & 0,0000 \\ 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$
[0,1]	$\begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0,1015 & 0,0014 & 0,0000 \\ 0,0014 & 0,0014 & 0,0000 \\ 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$
[0,1]	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2,2732 & 0,0166 & 0,0001 \\ 0,0166 & 0,0075 & 0,0000 \\ 0,0001 & 0,0000 & 0,0000 \end{bmatrix}$

Melalui penyelesaian persamaan Kalman *gain* pada **Persamaan (2.48)**, **Persamaan (2.51)**, dan **Persamaan (2.52)**, maka diperoleh parameter PID yang nilainya ditunjukkan pada **Tabel 3.13**.

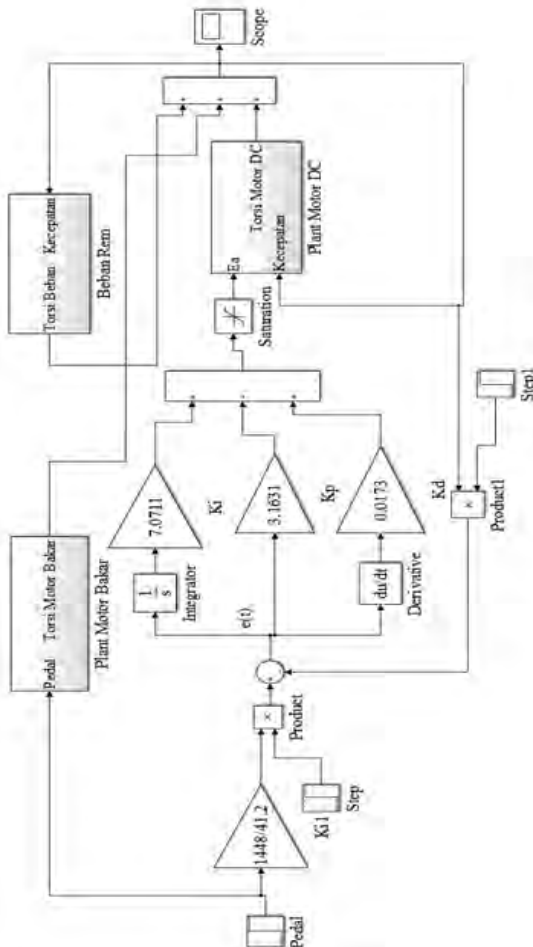
**Tabel 3.13** Nilai Parameter  $K_p$ ,  $K_i$ , dan  $K_d$

Parameter PID		
$K_p$	$K_i$	$K_d$
0,0742	0,2236	0,0009
0,2776	0,3162	0,0029
0,9635	1	0,0078
3,1631	7,0711	0,0173

Nilai Parameter  $K_p$ ,  $K_i$ , dan  $K_d$  yang telah didapatkan kemudian diuji untuk ditentukan nilai parameter PID mana yang paling baik untuk menyelesaikan permasalahan regulator pada sistem HEV.

### 3.7.1 Desain Simulink PID-LQR

Blok Simulink PID-LQR yang digunakan ditunjukkan pada Gambar 3.33.



Gambar 3.33 Desain Simulink PID-LQR

*--- Halaman ini sengaja dikosongkan ---*

## **BAB 4**

### **PENGUJIAN DAN ANALISIS**

#### **4.1 Gambaran Umum Pengujian Sistem**

Pengujian sistem dilakukan untuk mengetahui performansi kontroler yang sudah dirancang dalam menyelesaikan masalah pada sistem. Pengujian dilakukan dengan melihat pengaruh kinerja motor DC sebagai penggerak pembantu pada sistem *Hybrid Electric Vehicle*. Pada Tugas Akhir ini pengujian sistem dilakukan secara simulasi dan implementasi untuk melihat performansi dari *Hybrid Electric Vehicle* terhadap adanya beban berlebih berupa rem elektromagnetik. Pengujian dilakukan terhadap beberapa keadaan yaitu pada saat ICE diberikan beban tanpa bantuan motor DC dan pada saat ICE dan motor DC menanggung beban bersama menggunakan kontroler PID- *Linear Quadratic Regulator*.

#### **4.2 Simulasi Sistem**

Pada Tugas Akhir ini, simulasi dilakukan dengan menggunakan software MATLAB R2013a untuk mengetahui performansi sistem sebelum dan sesudah diberi kontroler.

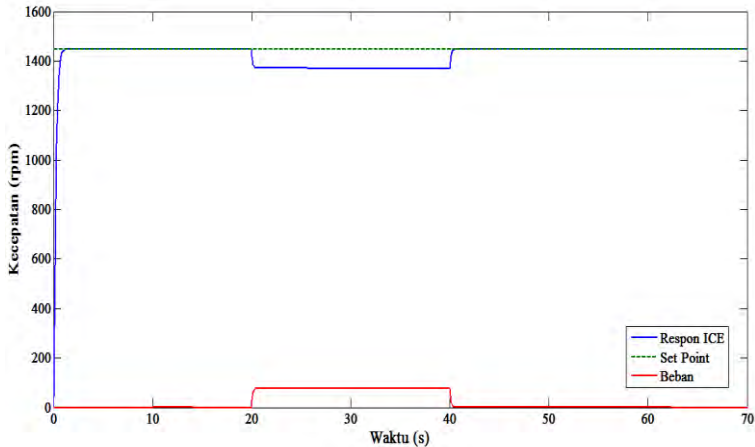
##### **4.2.1 Simulasi ICE Beban Nominal Tanpa Bantuan Motor Listrik**

Simulasi ini bertujuan untuk melihat respon ICE terhadap adanya efek beban. Pada proses pengujian, ICE akan diberi sinyal masukan berupa sinyal *step* sebesar 1448 rpm dengan diberi tiga macam beban transien yang berbeda. Klasifikasi beban untuk permasalahan regulator ini ditentukan dari arus yang diberikan pada rem elektromagnetik. Variasi beban yang diperbolehkan dalam pengujian kontroler harus sesuai dengan pemodelan nominal untuk ICE (disesuaikan dengan *range* pembebanan nominal), sehingga dibatasi beban yang diperbolehkan adalah pada *range* arus beban 0,56 A– 1,14 A.

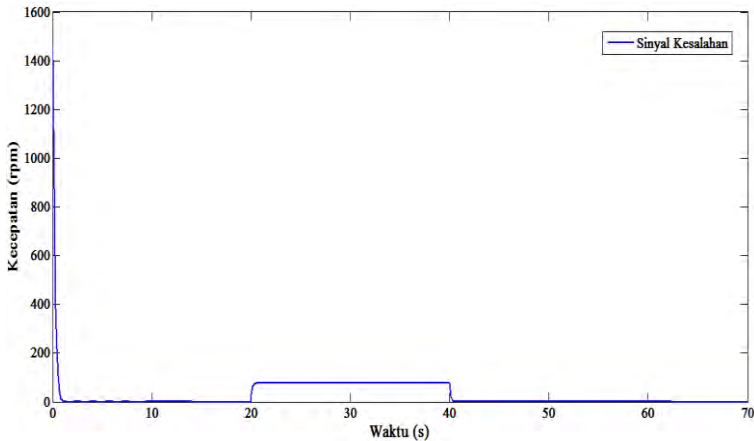
##### **a. Pengujian dengan Arus Beban 0,6 A**

Pada pengujian pertama dilakukan dengan arus rem sebesar 0,6 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 77,4 rpm. Hasil simulasi respon ICE saat diberi beban transien dapat dilihat pada **Gambar 4.1**. Pada **Gambar 4.1** dapat terlihat bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai

*steady state*. Sedangkan, sinyal kesalahan karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE. Sinyal kesalahan sistem dapat dilihat pada **Gambar 4.2**.



**Gambar 4.1** Respon Kecepatan Putar ICE dengan Arus Beban 0,6 A



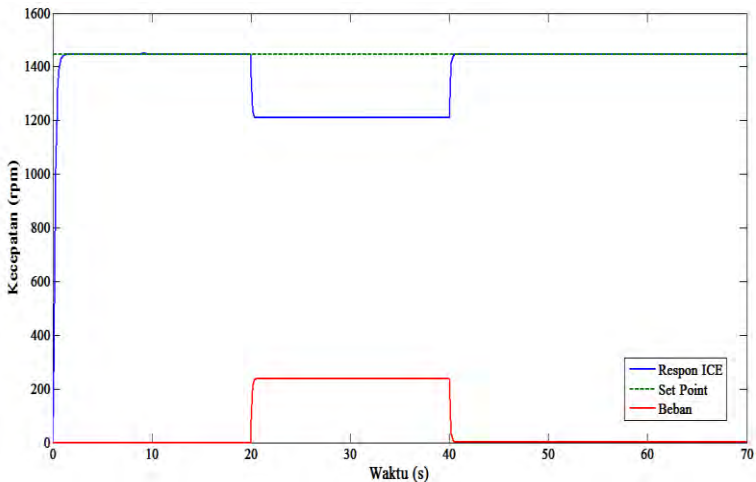
**Gambar 4.2** Sinyal Kesalahan ICE dengan Arus Beban 0,6 A

#### b. Pengujian dengan Arus Beban 0,7 A

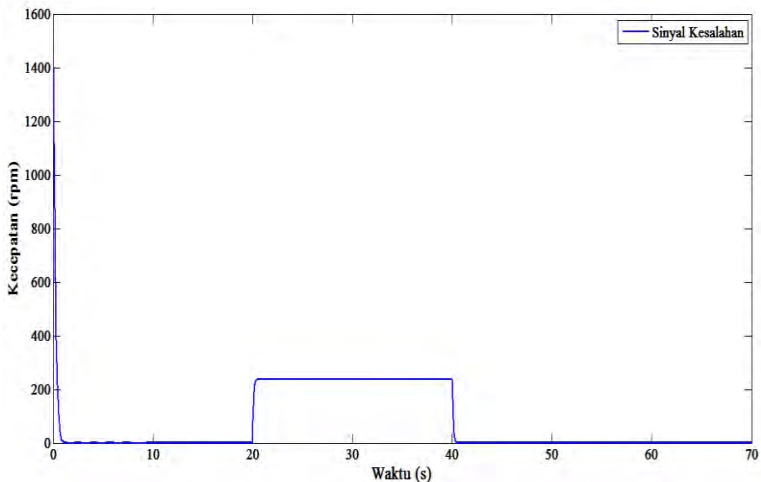
Pada pengujian ketiga dilakukan dengan memberikan arus rem sebesar 0,7 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 238 rpm. Hasil simulasi terlihat



pada **Gambar 4.3** bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai *steady state*. Sedangkan, sinyal kesalahan karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE. Sinyal kesalahan sistem dapat dilihat pada **Gambar 4.4**.



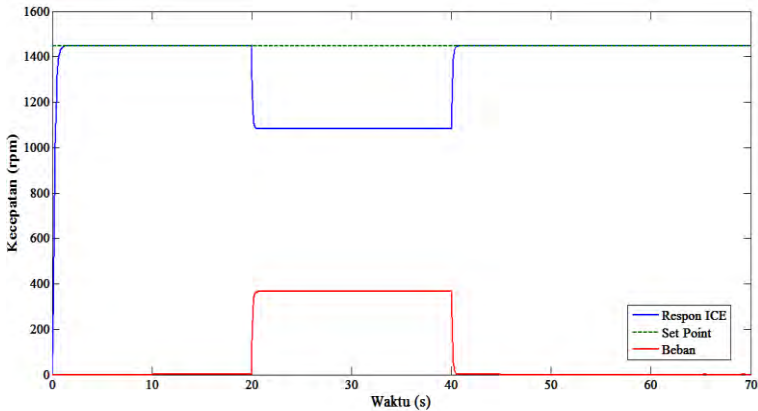
**Gambar 4.3** Respon Kecepatan Putar ICE dengan Arus Beban 0,7 A



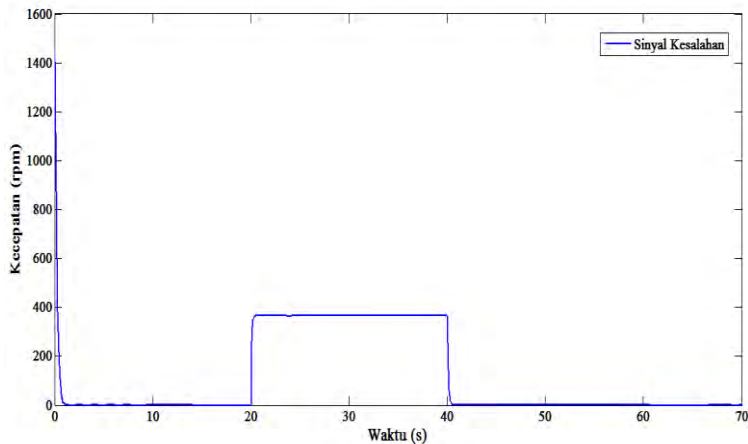
**Gambar 4.4** Sinyal Kesalahan ICE dengan Arus Beban 0,7 A

### c. Pengujian dengan Arus Beban 0,8 A

Pada pengujian kedua dilakukan dengan beban berupa arus rem sebesar 0,8 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 365 rpm. Hasil simulasi ditampilkan pada **Gambar 4.5** bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai *steady state*. Sedangkan, sinyal kesalahan pada **Gambar 4.6** karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE.



**Gambar 4.5** Respon Kecepatan Putar ICE dengan Arus Beban 0,8 A



**Gambar 4.6** Sinyal Kesalahan ICE dengan Arus Beban 0,8 A

Dari ketiga pengujian yang telah dilakukan dengan variasi beban yang berbeda dapat dilihat bahwa pada ketiga pengujian tersebut kesalahan akan bertambah saat terdapat beban dan kembali nol saat beban sudah tidak ada. Hal ini tidak sesuai dengan yang diharapkan, yaitu pada permasalahan regulator, kecepatan motor akan tetap stabil pada kondisi *steady state* mencapai kesalahan nol meskipun ICE diberi beban. Tetapi pada ketiga pengujian yang telah dilakukan, ICE saja tidak mampu digunakan mencapai kesalahan nol saat diberikan beban sehingga perlu diberikan tenaga penggerak pembantu yaitu motor DC.

#### 4.2.2 Simulasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID-Linear Quadratic Regulator

Simulasi ini bertujuan untuk melihat respon ICE dengan bantuan motor DC saat ada beban transien. Beban yang diberikan juga berupa arus rem terdapat 3 macam yaitu 0,6 A, 0,7, dan 0,8 A. Pengujian ini bertujuan untuk menyelesaikan permasalahan regulator dengan menggunakan kontroler PID-Linear Quadratic Regulator untuk mengontrol motor DC agar dapat memberikan bantuan berupa torsi kepada ICE.

##### 4.2.2.1 Model Simulasi

Pemilihan parameter PID untuk mempertahankan kecepatan pada permasalahan regulator berdasarkan nilai kesalahan terkecil dilihat dari nilai IAE (*Integral Absolute Error*) yang terkecil untuk setiap pembebanan. Berdasarkan hasil simulasi (simulink MATLAB dapat dilihat pada **Lampiran D**) didapatkan nilai IAE saat arus beban 0,6 A, 0,7 A, dan 0,8 A pada **Tabel 4.1**, **Tabel 4.2**, dan **Tabel 4.3**. Pada **Tabel 4.1**, **Tabel 4.2**, dan **Tabel 4.3** dapat dilihat bahwa nilai IAE terkecil didapat dengan parameter  $K_p = 3,1631$ ,  $K_i = 7,0711$ , dan  $K_d = 0,0173$  dengan matriks

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -635,9768 & -76,4785 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ -12295,58 \end{bmatrix}$$

$$R = [0,1], Q = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, P = \begin{bmatrix} 2,2372 & 0,0166 & 0,0001 \\ 0,0166 & 0,0075 & 0,0000 \\ 0,0001 & 0,0000 & 0,0000 \end{bmatrix}$$

sehingga parameter PID tersebut dianggap paling baik untuk dapat membantu kinerja ICE memperbaiki kinerja HEV mencapai *set point* ketika diberi beban.

**Tabel 4.1** Nilai IAE Saat Arus Beban 0,6 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[0,1]	$\begin{bmatrix} 0,005 & 0 & 0 \\ 0 & 0,001 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,0742	0,2236	0,0009	155,1	$3,878 \times 10^4$
[0,1]	$\begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,2776	0,3162	0,0029	108,8	$2,097 \times 10^5$
[0,1]	$\begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,9635	1	0,0078	33,76	$2,136 \times 10^6$
[0,1]	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	3,1631	7,0711	0,0173	4,773	$2,286 \times 10^7$

**Tabel 4.2** Nilai IAE Saat Arus Beban 0,7 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[0,1]	$\begin{bmatrix} 0,005 & 0 & 0 \\ 0 & 0,001 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,0742	0,2236	0,0009	293,4	$1,018 \times 10^5$
[0,1]	$\begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,2776	0,3162	0,0029	206,6	$2,752 \times 10^5$
[0,1]	$\begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,9635	1	0,0078	64,69	$2,211 \times 10^6$
[0,1]	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	3,1631	7,0711	0,0173	9,149	$2,298 \times 10^7$

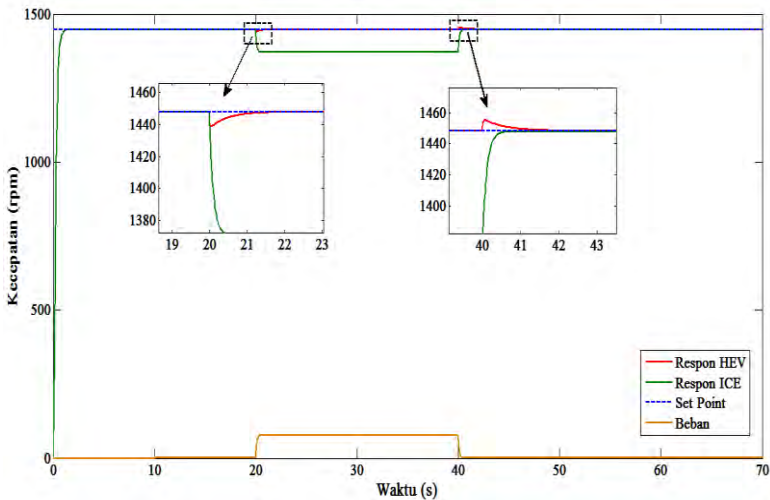
**Tabel 4.3** Nilai IAE Saat Beban 0.8 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[0,1]	$\begin{bmatrix} 0,005 & 0 & 0 \\ 0 & 0,001 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,0742	0,2236	0,0009	385,4	$1,648 \times 10^5$
[0,1]	$\begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,2776	0,3162	0,0029	277,1	$3,422 \times 10^5$
[0,1]	$\begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0,1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	0,9635	1	0,0078	85,25	$2,292 \times 10^6$
[0,1]	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	3,1631	7,0711	0,0173	12,06	$2,314 \times 10^7$

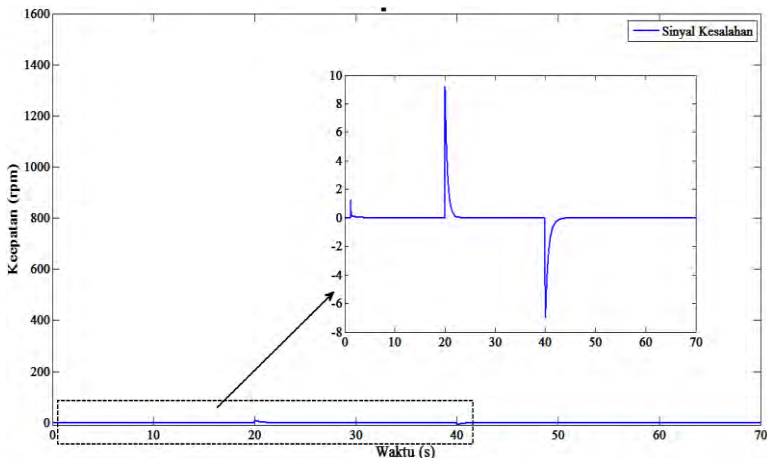
**a. Pengujian dengan Arus Beban 0,6 A**

Pengujian pertama dilakukan dengan memberikan arus beban pada rem elektromagnetik sebesar 0,6 A sehingga terjadi penurunan kecepatan putar sebesar 77,4 rpm. Ketika terjadi penurunan kecepatan, motor DC dengan menggunakan kontroler PID-LQR dapat membantu kinerja ICE untuk memperbaiki performansinya mencapai *set point* dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 9 rpm. seperti yang terlihat pada **Gambar 4.7**. Sinyal kesalahan dari keseluruhan sistem terhadap *set point* ketika terdapat beban seperti yang ditampilkan pada **Gambar 4.8**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Pada **Gambar 4.8** dapat dilihat bahwa motor DC menggunakan kontroler PID-LQR mampu mencapai kesalahan hingga mendekati nol dengan menghilangkan kesalahan sebesar 99,38 % ketika sistem diberikan beban. Sinyal kontrol dapat dilihat pada **Gambar 4.9** dimana sinyal kontrol dikondisikan aktif ketika ICE berada pada kondisi *steady state* yaitu pada detik ke-1,2. Pada Tugas Akhir ini, ICE dianggap telah mencapai kondisi *steady state* dan pemberian beban menyebabkan penurunan kecepatan pada ICE. Ketika ICE telah mencapai *set point*,

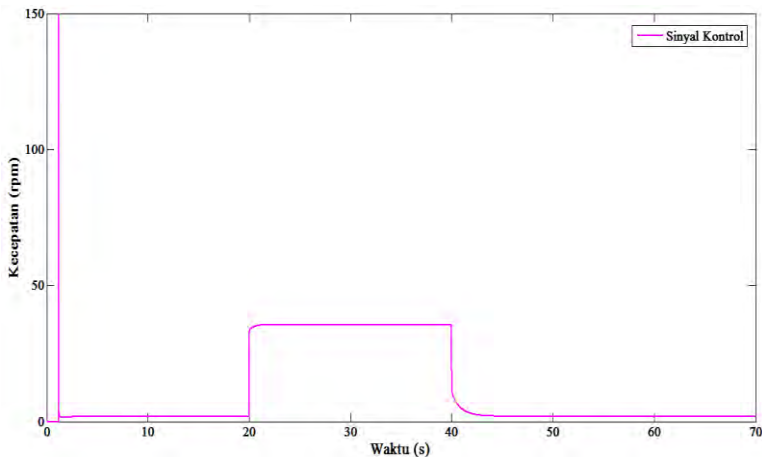
tidak terdapat kesalahan antara sinyal keluaran sistem dengan *set point* sehingga sinyal kontrol tidak bekerja. Ketika sistem diberikan beban pada selang waktu tertentu terjadi penurunan kecepatan sehingga terdapat kesalahan antara sinyal keluaran sistem dengan *set point*. Saat terjadi kesalahan maka sinyal kontrol bekerja untuk menghilangkan kesalahan yang terjadi. Saat beban dilepaskan, ICE kembali lagi pada kondisi *steady state* dan sinyal kontrol tidak lagi bekerja, sehingga sinyal kontrol hanya akan bekerja ketika sistem diberikan beban. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*. Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator.



**Gambar 4.7** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,6



**Gambar 4.8** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,6 A

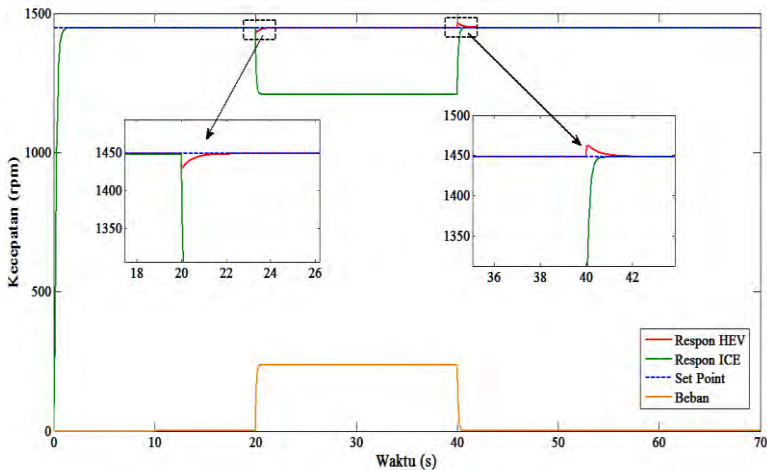


**Gambar 4.9** Sinyal Kontrol Motor DC Saat Arus Beban 0,6 A

#### **b. Pengujian dengan Arus Beban 0,7 A**

Pengujian kedua dilakukan dengan memberikan arus rem yang lebih tinggi yaitu sebesar 0,7 A. Karena diberikan beban yang lebih tinggi, maka penurunan kecepatan juga lebih tinggi yaitu sebesar 238 rpm. Pada **Gambar 4.10** dapat terlihat, dengan kontroler PID-LQR, ICE dan motor DC mampu menanggung beban bersama untuk

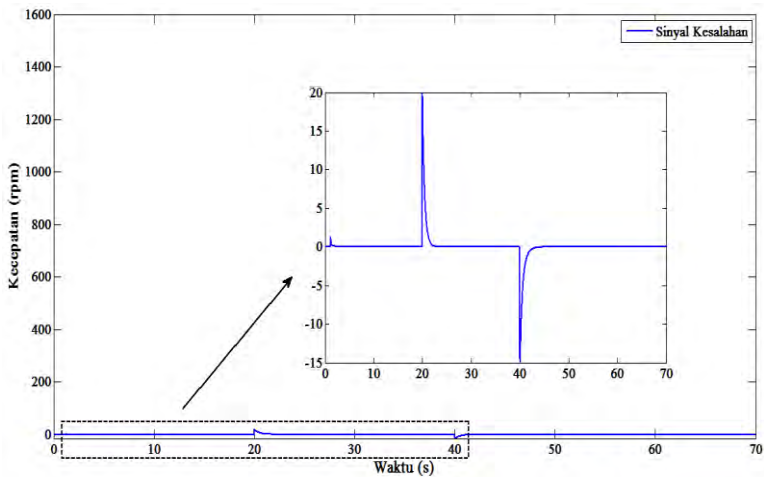
mempertahankan kecepatan menuju nilai *steady state* dari ICE dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 20 rpm. Sinyal kesalahan dari keseluruhan sisem ditampilkan pada **Gambar 4.11**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Motor DC menggunakan kontroler PID-LQR mampu menghilangkan kesalahan sebesar 99,69 % ketika terjadi pembebanan. Hal ini menandakan bahwa kontroler PID-LQR mampu bekerja dengan baik sehingga mampu mencapai kesalahan hingga mendekati nol.. Sinyal kontrol dapat dilihat pada **Gambar 4.12**. Pada Tugas Akhir ini, ICE dianggap telah mencapai kondisi *steady state* dan pemberian beban menyebabkan penurunan kecepatan pada ICE. Ketika ICE telah mencapai *set point*, sinyal kontrol tidak bekerja, artinya tidak terdapat kesalahan antara sinyal keluaran sistem dengan *set point*. Ketika sistem diberikan beban pada selang waktu tertentu, sinyal kontrol bekerja hingga beban dilepaskan. Saat beban dilepaskan, ICE kembali lagi pada kondisi *steady state* dan sinyal kontrol tidak lagi bekerja. Sehingga sinyal kontrol hanya akan bekerja ketika sistem diberikan beban. Pada pemberian beban yang lebih tinggi sinyal kontrol yang bekerja juga akan lebih besar untuk mengurangi kesalahan sistem.



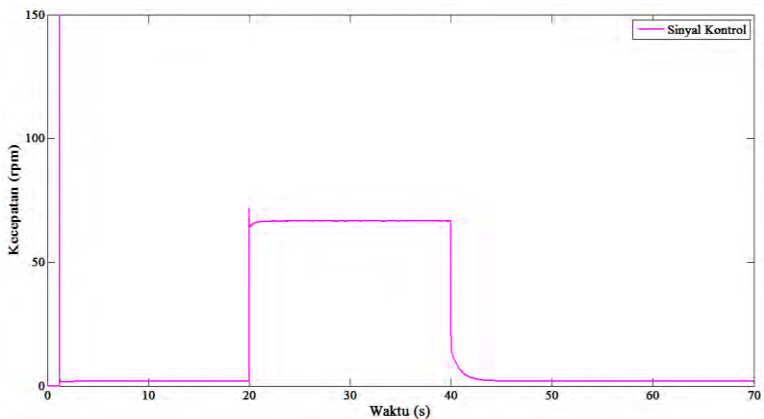
**Gambar 4.10** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,7 A



Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*.



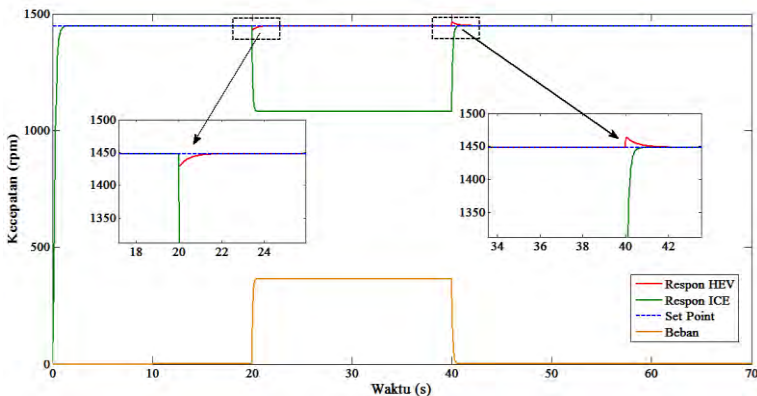
**Gambar 4.11** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,7 A



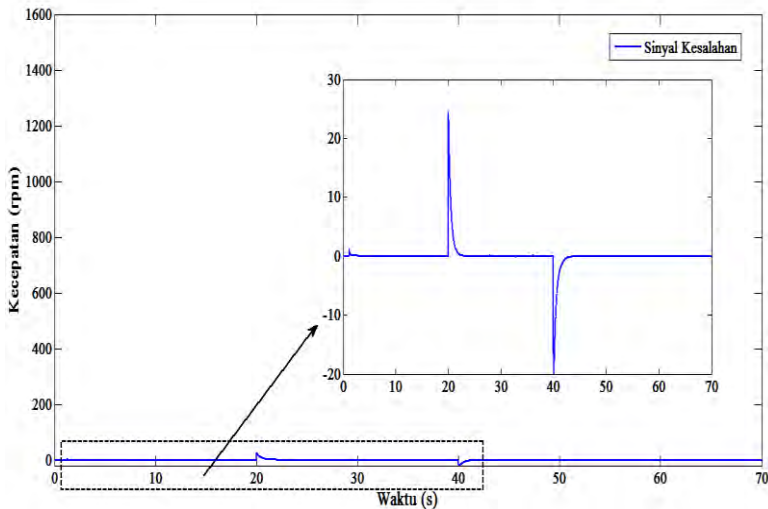
**Gambar 4.12** Sinyal Kontrol Motor DC Saat Arus Beban 0,7 A

### c. Pengujian dengan Arus Beban 0,8 A

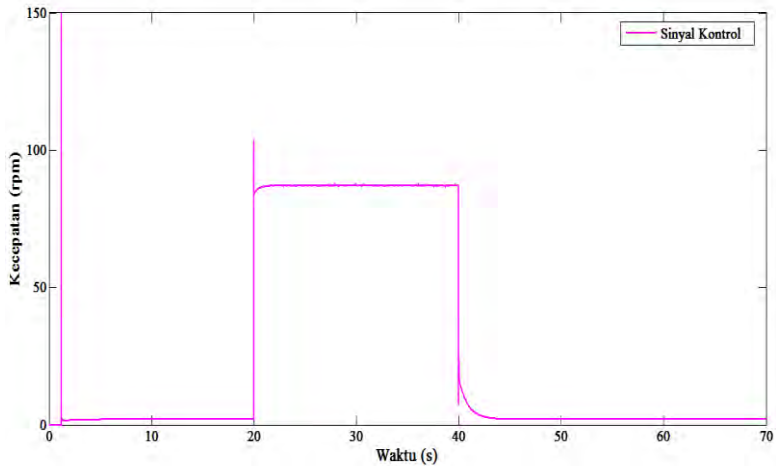
Pengujian kedua dilakukan dengan memberikan arus yang lebih tinggi pada rem elektromagnetik sebesar 0,8 A sehingga terjadi penurunan kecepatan putar sebesar 365 rpm. Dengan kontroler PID-LQR, ICE dan motor DC mampu menanggung beban bersama untuk mempertahankan kecepatan menuju nilai *steady state* dari ICE seperti yang terlihat pada **Gambar 4.13** dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 25 rpm. Sinyal kesalahan dari keseluruhan sistem terhadap *set point* ketika terdapat beban seperti yang ditampilkan pada **Gambar 4.14**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Motor DC menggunakan kontroler PID-LQR mampu menghilangkan kesalahan sebesar 99,80 %. Sinyal kontrol dapat dilihat pada **Gambar 4.15**. Pada pemberian beban yang lebih tinggi lagi sinyal kontrol yang bekerja juga akan lebih besar untuk mengurangi kesalahan sistem. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*. Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator.



**Gambar 4.13** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,8 A



**Gambar 4.14** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,8 A



**Gambar 4.15** Sinyal Kontrol Motor DC Saat Arus Beban 0,8 A

Dari ketiga pengujian yang telah dilakukan dengan variasi beban yang berbeda, motor DC dengan menggunakan kontroler PID-LQR dapat membantu kinerja ICE untuk memperbaiki performansinya mencapai *set point* saat ada beban. Salah satu tugas kontroler adalah

meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal *set point* dan sinyal aktual. Berdasarkan hasil pengujian secara simulasi, perancangan kontroler PID-LQR untuk mengatur motor DC dalam membantu ICE sudah sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang senantiasa (diinginkan) sama dengan sinyal *set point*.

Jika dilihat dari analisis kestabilannya, *eigen value* dari ketiga pengujian ini memiliki nilai kurang dari nol atau berada pada sebelah kiri sumbu imajiner sehingga dapat disimpulkan bahwa respon stabil. Hal ini menandakan bahwa kontroler PID-LQR memiliki performansi yang baik untuk permasalahan regulasi sehingga dapat mengatur motor DC untuk membantu kinerja ICE mempertahankan kondisi *steady state* nya.

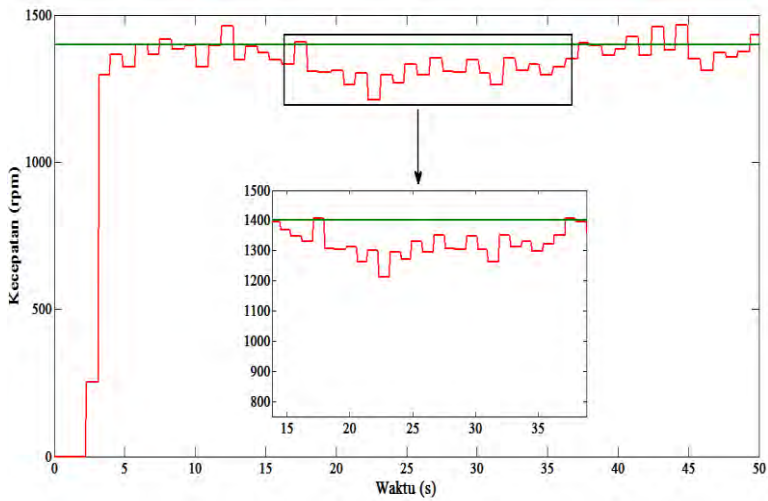
### 4.3 Implementasi Sistem

Implementasi sistem pengaturan kecepatan HEV dilakukan pada *plant* secara langsung. Pengujian dilakukan dengan menghubungkan *plant* HEV terhadap kontroler yang telah dirancang pada MATLAB R2013a melalui Arduino *board* dan *driver* motor DC. Pembacaan *rotary encoder* dan pedal masuk pada *analog read* Arduino dan menjadi *set point*. Sinyal kontrol dari MATLAB dikirim melalui pengiriman data serial ke *driver* motor DC untuk menggerakkan motor DC sesuai pengaturan PWM nya. Pengujian pada sistem dilakukan dengan memberikan variasi beban berupa tegangan rem nominal sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC.

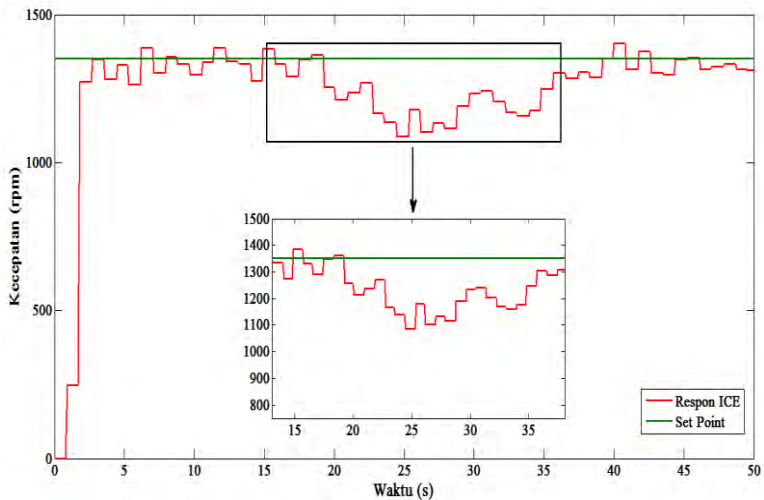
#### 4.3.1 Implementasi ICE Beban Nominal Tanpa Bantuan Motor Listrik

Pada pengujian ini, hanya dilihat respon ICE terhadap adanya pemberian tegangan sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC pada rem elektromagnetik yang dapat menyebabkan penurunan kecepatan pada ICE. Respon ICE dapat dilihat pada **Gambar 4.16** dan **Gambar 4.17**.

Pada **Gambar 4.16** dan **Gambar 4.17** dapat dilihat bahwa pada saat rem elektromagnetik diberi tegangan sebesar 50 VDC, dan 57,15 VDC, maka kecepatan ICE akan menurun. Semakin besar tegangan yang diberikan pada rem elektromagnetik, maka semakin besar pula penurunan kecepatan yang terjadi pada ICE.



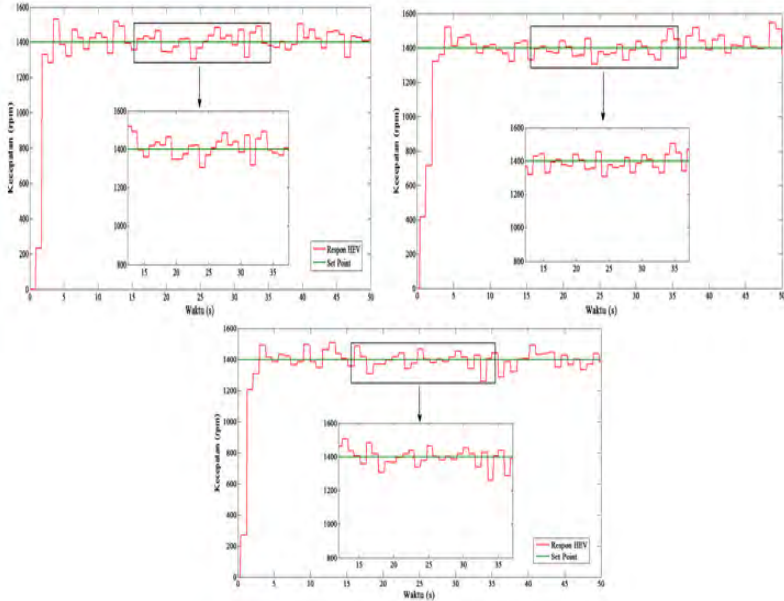
**Gambar 4.16** Respon Kecepatan Putar ICE Hasil Impementasi dengan Tegangan Rem 50 VDC



**Gambar 4.17** Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 57,15 VDC

#### 4.3.2 Implementasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID-Linear Quadratic Regulator

Pada pengujian ini, ICE sebagai penggerak utama dibantu oleh motor DC sebagai penggerak pembantu. Setelah itu diberikan tegangan pada rem elektromagnetik sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC sebagai efek beban yang dapat menyebabkan penurunan kecepatan pada ICE. Respon HEV secara keseluruhan dapat dilihat pada **Gambar 4.18**.



**Gambar 4.18** Respon Kecepatan Putar HEV Hasil Implementasi Menggunakan Kontroler PID-LQR dengan 3 Variasi Beban

Pada **Gambar 4.18** dapat dilihat bahwa pada pemberian 3 variasi beban berupa tegangan rem, motor DC menggunakan kontroler PID-LQR mampu membantu mesin bakar untuk mengembalikan kecepatan hingga mendekati *set point* pada saat terjadi pembebanan meskipun masih terdapat  $e_{ss}$  sebesar 20,67 % ketika pemberian tegangan rem 42,86 VDC,  $e_{ss}$  sebesar 28,8 % ketika pemberian tegangan rem 50 VDC, dan  $e_{ss}$  sebesar 26,6 % ketika pemberian tegangan rem 57,15 VDC.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan analisis, didapatkan kesimpulan sebagai berikut :

1. Parameter kontroler PID berupa  $K_p$ ,  $K_i$ , dan  $K_d$  yang optimal dapat dicari menggunakan pendekatan *Linear Quadratic Regulator* berdasarkan penerapan aturan tertentu yang sesuai.
2. Nilai parameter  $K_p = 3,163$ ,  $K_i = 7,0711$ , dan  $K_d = 0,0173$  merupakan nilai parameter PID terbaik untuk dapat membantu kinerja ICE memperbaiki performansinya mencapai *set point* ketika diberi beban.
3. Hasil pengujian secara simulasi menggunakan kontroler PID-LQR, motor DC mampu membantu kerja ICE pada saat terdapat beban sehingga kecepatan yang diinginkan dapat dipertahankan sehingga dapat menghilangkan kesalahan sebesar 99,38 % ketika sistem diberikan arus beban 0,6 A, 99,69 % ketika sistem diberikan arus beban 0,7 A, dan 99,80 % saat sistem diberikan arus beban sebesar 0,8 A.
4. Pada hasil implementasi, motor DC juga mampu membantu ICE sehingga kecepatan putar HEV mendekati *set point*, meskipun masih terdapat *error steady state* hingga 28,8 %.

#### **5.2 Saran**

Pada Tugas Akhir ini, pemilihan matriks **Q** dan **R** untuk memperoleh parameter PID masih menggunakan metode *trial and error*. Diharapkan pada penelitian selanjutnya, pemilihan matriks **Q** dan **R** dapat menggunakan metode komputasi cerdas agar mendapatkan hasil yang lebih optimal. Selain itu, identifikasi dan pemahaman tentang *plant* yang digunakan dapat membantu perancangan sistem yang lebih baik.

*--- Halaman ini sengaja dikosongkan ---*



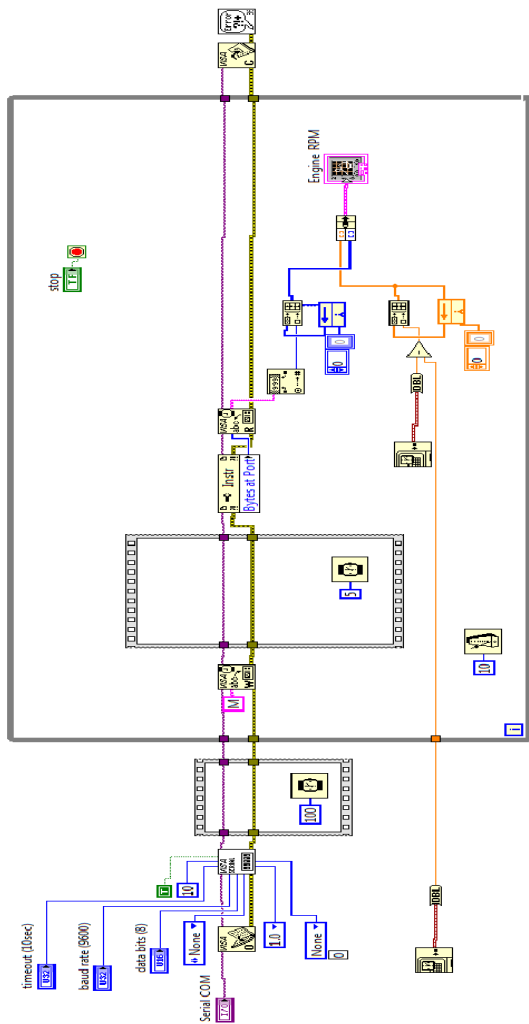
## DAFTAR LAMPIRAN

<b>Lampiran A</b>	Blok LabView untuk Identifikasi Statis .....	107
<b>Lampiran B</b>	Diagram Simulink Sistem HEV .....	108
<b>Lampiran C</b>	Diagram Simulink <i>Plant</i> Motor DC .....	109
<b>Lampiran D</b>	Diagram Simulink Pencarian Nilai IAE .....	110
<b>Lampiran E</b>	Diagram Simulink Implementasi Sistem HEV .....	111
<b>Lampiran F</b>	Program MATLAB .....	112
<b>Lampiran G</b>	Program Arduino untuk Pembacaan Kecepatan Motor DC .....	113
<b>Lampiran H</b>	Program Arduino untuk Implementasi .....	114
<b>Lampiran I</b>	Program Perhitungan RMSE .....	115

*--- Halaman ini sengaja dikosongkan---*

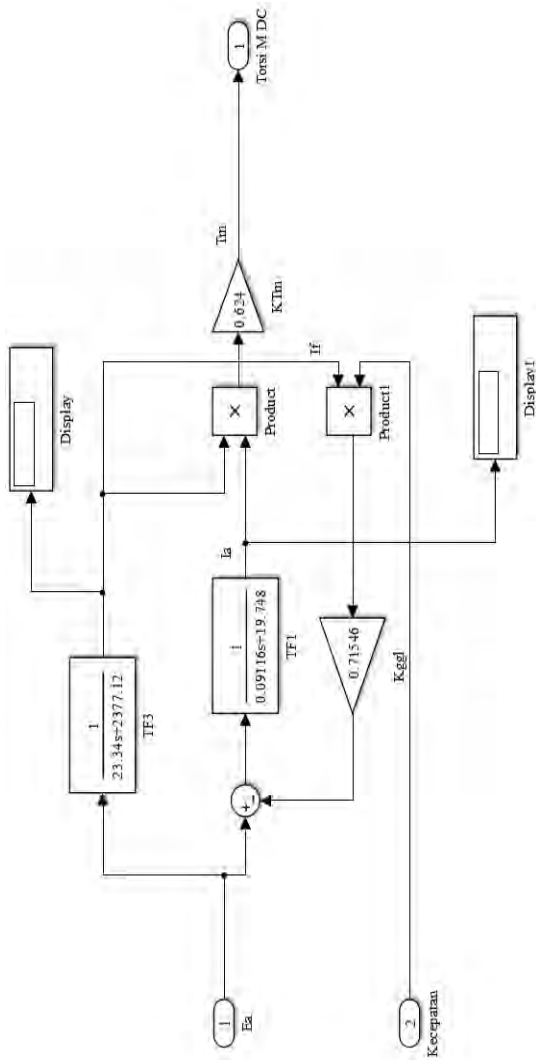
## LAMPIRAN

### A. Blok LabView untuk Identifikasi Statis

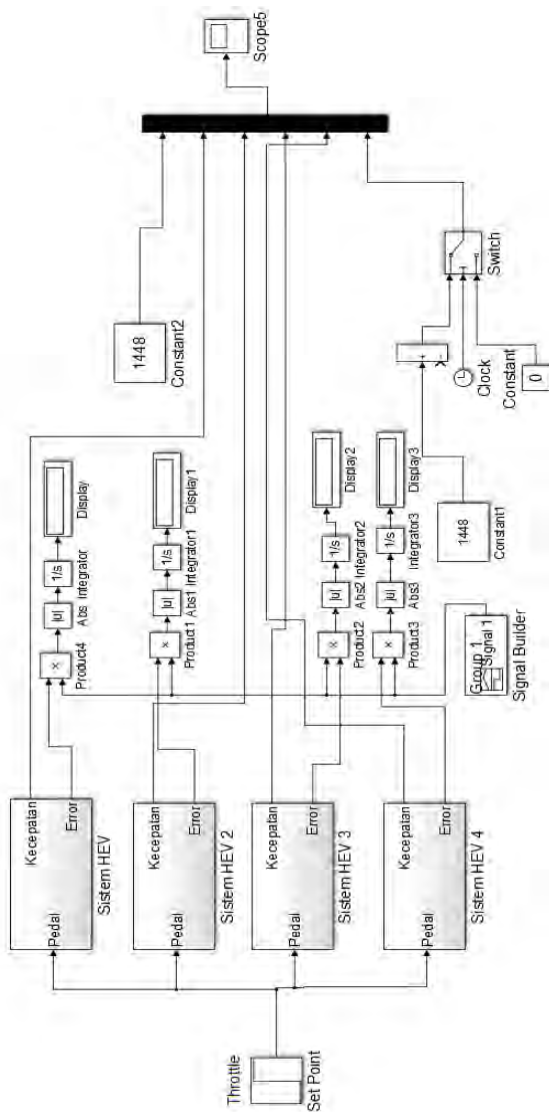




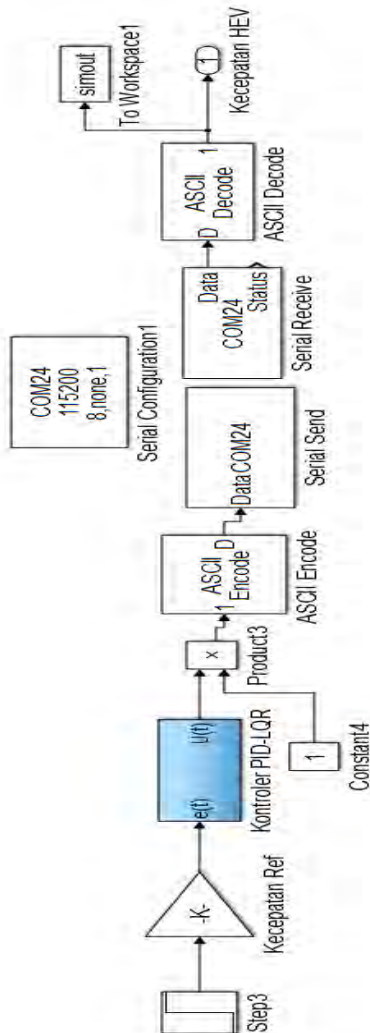
### C. Diagram Simulink *Plant* Motor DC



D. Diagram Simulink Pencarian nilai IAE



## E. Diagram Simulink Implementasi Sistem HEV



## **F. Program MATLAB**

```
% Judul Tugas Akhir :  
% Pengaturan Kecepatan pada Simulator Paralel Hybrid Electric Vehicle  
% Menggunakan Metode PID-Linear Quadratic Regulator  
%  
% Fannieshe Hamada  
% 2211100207  
%  
% Jurusan Teknik Elektro ITS  
% Surabaya 2015  
  
% Parameter PID-LQR  
A = [0 1 0; 0 0 1; 0 -635.9768 -76.4785]; % Matriks A  
B = [0; 0; -12295.58]; % Matriks B  
Q = [50 0 0; 0 1 0; 0 0 0]; % Matriks Pembobot Q  
R = [1]; % Matriks Pembobot R  
  
% Perhitungan parameter Ki, Kp, dan Kd  
[K,P,EV]= lqr(A,B,Q,R)  
% P = Riccati Matrix  
% EV = Eigen Value
```



## **G. Program Arduino untuk Pembacaan Kecepatan Motor DC**

```
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(encoder, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    delay(1);
    if (Serial.read() == 'M')
    {
      while (Serial.available())
      {
        Serial.read();
      }
      unsigned long waktu = pulseIn(encoder, HIGH);
      float jeda = waktu;
      jeda = jeda / 1000000;
      float kecepatan = (1 / (jeda * 72)) * 60;
      float y = 229.9536927+(0.680455391*kecepatan);
      word rpm = y;
      Serial.print(rpm);
    }
  }
}
```

## H. Program Arduino Untuk Implementasi

```
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;
char disp[1];
int sinyalkontrol;
int sinyalkontrolarduino;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(encoder, INPUT);
    pinMode(13, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    if (Serial.available() > 0)
        {Serial.parseInt();}
    delay(1);
    unsigned long waktu = pulseIn(encoder, HIGH);
    float jeda = waktu;
    jeda = jeda / 1000000;
    float kecepatan = (1 / (jeda * 72)) * 60;
    float y = 229.9536927 + (0.680455391 * kecepatan);
    word rpm = y;
    sprintf(disp, "%4d", rpm);
    Serial.print(disp);

    if (Serial.available() > 0) {
        sinyalkontrol = Serial.parseInt();
        sinyalkontrolarduino = map(sinyalkontrol, 0, 150, 0, 1000);
        digitalWrite(13, LOW);
    }
}
```

```

    delay(sinyalkontrolarduino);
    digitalWrite(13,HIGH);
    delay(1000-sinyalkontrolarduino);
  }
}

```

## I. Program Perhitungan RMSE

```

clc;
n=size(xxx,1);
A=xxx(:,2);
M=xxx(:,3);
sumei=0;
for i=1:n;
    ei=(A(i)-M(i))/A(i)
    sumei=sumei+ei^2
end
sumei
n
RMSE=sqrt(sumei/n)

```

*--- Halaman ini sengaja dikosongkan ---*



**FINAL PROJECT - TE 141599**

**VELOCITY CONTROL OF PARALLEL HYBRID ELECTRIC  
VEHICLE SIMULATOR USING PID - LINEAR QUADRATIC  
REGULATOR METHOD**

Fanniesha Hamada  
NRP 2211100207

Advisor  
Ir. Rusdhianto Effendie A.K., MT.  
Ir. Ali Fatoni, MT.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

# **PENGATURAN KECEPATAN PADA SIMULATOR *PARALLEL HYBRID ELECTRIC VEHICLE* MENGGUNAKAN METODE *PID - LINEAR QUADRATIC REGULATOR***

Fanniesha Hamada – 2211100207

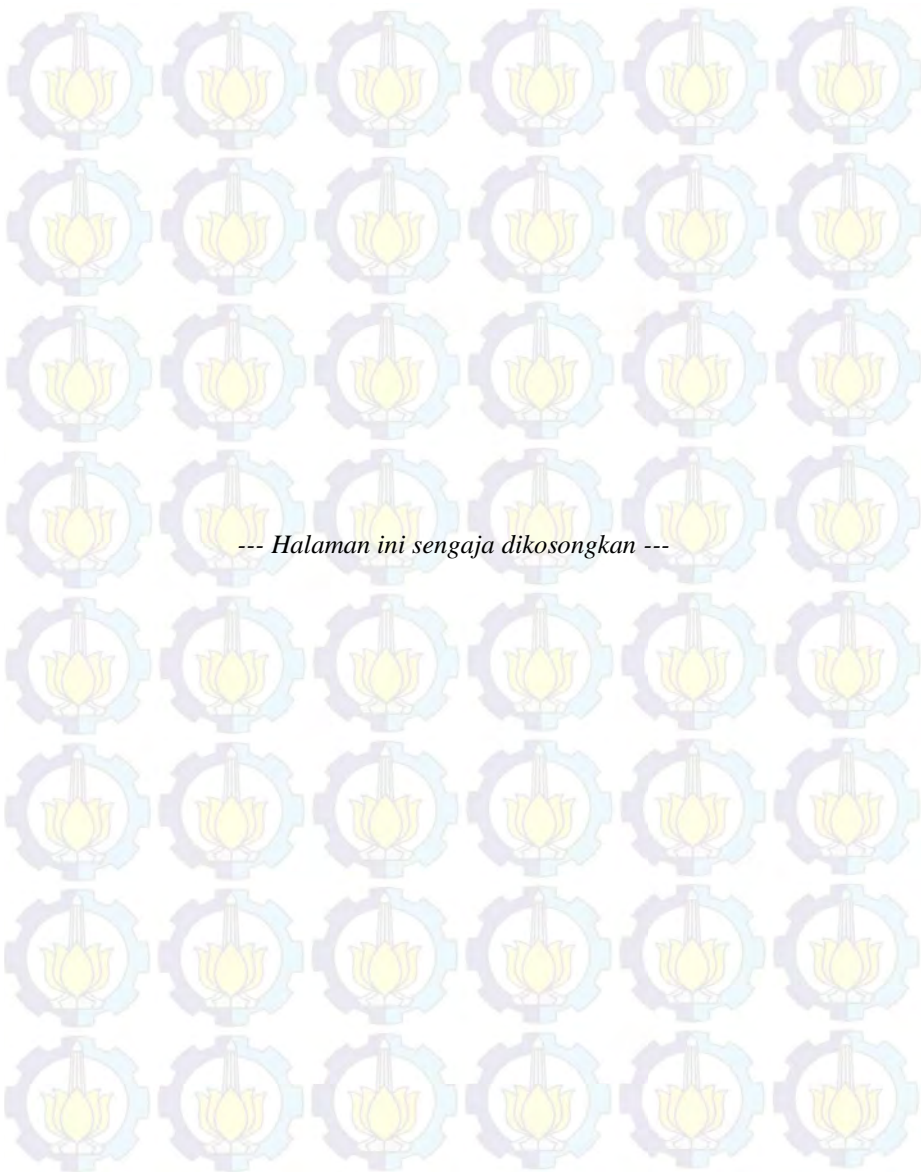
Pembimbing I : Ir. Rushdianto Effendie A.K., MT.

Pembimbing II : Ir. Ali Fatoni MT.

## **ABSTRAK**

*Hybrid Electric Vehicle* (HEV) merupakan suatu kendaraan dengan konsep ramah lingkungan dan hemat energi yang diharapkan menjadi salah satu alternatif menanggulangi efek rumah kaca dan krisis energi. HEV menggabungkan kinerja *Internal Combustion Engine* (ICE) atau mesin bakar dan motor listrik. Pada HEV dengan konfigurasi paralel, ICE dan motor listrik dapat bekerja bersama-sama. Pada Tugas Akhir ini digunakan Simulator *Parallel Hybrid Electric Vehicle* (PHEV). Simulator ini merepresentasikan kondisi nyata HEV namun dalam skala yang lebih kecil. Simulator ini terdiri dari mesin bakar 2 tak sebagai penggerak utama, motor DC sebagai penggerak pembantu, dan beban berupa rem magnetik arus *eddy*. Ketika terjadi permasalahan regulator akibat pembebanan lebih pada kendaraan, kecepatan putar pada ICE menurun sehingga kecepatan HEV tidak sesuai dengan *output* yang diharapkan. Oleh karena itu, dibutuhkan suatu kontroler untuk melakukan pengaturan kerja dari motor listrik agar bekerja sesuai dengan kebutuhan yang diinginkan. Kontroler PID – *Linear Quadratic Regulator* (PID-LQR) digunakan untuk memperbaiki performansi kerja HEV agar mampu membantu ICE mencapai kecepatan putar yang seharusnya. Berdasarkan hasil pengujian secara simulasi didapatkan motor DC menggunakan kontroler PID-LQR mampu membantu kinerja ICE sehingga dapat mengembalikan respon sistem menuju nilai *steady state* ketika terjadi pembebanan berlebih pada rentang arus beban rem nominal 0.56-1.14 A. Berdasarkan hasil pengujian secara implementasi, motor DC dapat membantu kinerja ICE, namun masih terdapat *error steady state*.

**Kata Kunci** : *Hybrid Electric Vehicle*, ICE, Motor Listrik, Motor DC, PID-LQR



# **VELOCITY CONTROL OF PARALLEL HYBRID ELECTRIC VEHICLE SIMULATOR USING PID - LINEAR QUADRATIC REGULATOR METHOD**

Fanniesha Hamada – 2211100207

Advisor I : Ir. Rushdianto Effendie A.K., MT.

Advisor II : Ir. Ali Fatoni MT.

## **ABSTRACT**

*Hybrid Electric Vehicle (HEV) is a vehicle with the concept of environmentally friendly and energy saving which is expected to be an alternative to combat the greenhouse effect and energy crisis. HEV combines the performance of Internal Combustion Engine (ICE) and electric motor. In the parallel HEV configuration, ICE and electric motor can work together. Parallel Hybrid Electric Vehicle simulator is used in this final project. This simulator represents real condition of HEV but in a smaller scale. This simulator consist of a stroke-2 combustion engine as the prime mover, DC motor as an assist mover, and the load of magnetic eddy current brake. When there is excessive load on a vehicle called regulator problem, the rotational speed decreases so that the rotational speed of HEV doesn't match with the expected output. Therefore, a controller is needed for working arrangement of the electric motor according to the desired needs. PID – Linear Quadratic Regulator Controller (PID-LQR) is used to improve work performance of HEV in order to assist the combustion engine to achieve an appropriate rotating speed. Based on simulation testing result, DC motor with PID-LQR controler can help the performance of ICE so that it can restore the system response towards steady state value at excessive load in the range of 0,56-1,14 A of nominal load current brake. Based on implementation testing result, DC motor can assist the combustion engine, however there's still have an error steady state.*

**Keywords :** Hybrid Electric Vehicle, Internal Combustion Engine, Electric Motor, DC Motor, PID-LQR





## KATA PENGANTAR

Alhamdulillah penulis ucapkan atas kehadiran Allah SWT karena atas segala rahmat dan hidayah-Nya, Tugas Akhir ini dapat terselesaikan dengan baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada bidang studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul :

### **PENGATURAN KECEPATAN PADA SIMULATOR PARALLEL HYBRID ELECTRIC VEHICLE MENGGUNAKAN METODE PID - LINEAR QUADRATIC REGULATOR**

Tugas Akhir ini disusun berdasarkan pembelajaran dan pengujian yang telah dilakukan dan tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Kedua orang tua penulis yang selalu memberi dukungan penuh dalam bentuk apapun.
2. Dosen pembimbing, Bapak Ir. Rusdhianto Effendie A.K., MT. dan Ir. Ali Fatoni, MT yang selalu membimbing dan memberikan banyak ilmu dan motivasi.
3. Rekan satu tim penulis, Ajib Setiawan N., Sentosa Sondang O., Alif Ridwan M, Aulia Rahma A, dan Fahrul.
4. Angkatan e-51 yang telah memberi dukungan moril dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa buku Tugas Akhir ini belum sempurna, oleh karena itu penulis sangat mengharapkan saran dan masukan untuk perbaikan di masa yang akan datang. Semoga buku Tugas Akhir ini dapat bermanfaat dalam pengembangan ilmu dan teknologi dan juga bagi pembaca di kemudian hari.

Surabaya, 30 Juni 2015

Fanniesha Hamada  
2211100207



## DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
HALAMAN PENGESAHAN.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
1.6 Relevansi .....	3
BAB 2 DASAR TEORI .....	5
2.1 <i>Hybrid Electric Vehicle</i> (HEV) .....	5
2.1.1 HEV Konfigurasi Paralel .....	5
2.1.2 HEV Konfigurasi Seri .....	6
2.1.3 HEV Konfigurasi Seri-Paralel .....	7
2.2 Simulator <i>Hybrid Electric Vehicle</i> .....	9
2.2.1 <i>Internal Combustion Engine</i> (ICE) 2 Tak .....	9
2.2.2 Motor Listrik DC.....	12
2.2.3 Rem Elektromagnetik.....	16
2.2.4 <i>Timing Belt</i> .....	18
2.2.5 <i>Rotary Encoder</i> .....	22
2.3 Sistem Minimum Arduino Uno R3 .....	22
2.4 Identifikasi Sistem .....	24
2.4.1 Identifikasi Strejc .....	25
2.4.2 Identifikasi Fisik .....	27
2.5 Representasi <i>State Space</i> .....	30
2.6 <i>Root Mean Square Error</i> .....	34
2.7 Kontroler .....	34
2.7.1 Kontroler <i>Proportional Integral Derivative</i> (PID) ...	35
2.7.2 Kontroler Optimal LQR .....	38
2.7.3 Kontroler PID-LQR .....	38



BAB 3 PERANCANGAN SISTEM.....	41
3.1 Gambaran Umum Sistem .....	41
3.2 Perancangan Perangkat Keras .....	43
3.2.1 Perancangan Perangkat Mekanik .....	44
3.2.2 Perancangan Perangkat Elektronik.....	51
3.3 Perancangan Perangkat Lunak .....	55
3.3.1 <i>Software</i> Arduino .....	55
3.3.2 <i>Software</i> LabView.....	56
3.3.3 <i>Software</i> MATLAB.....	57
3.4 Pengujian Sistem.....	58
3.4.1 Pengujian Sensor.....	59
3.5 Proses Identifikasi dan Pemodelan Sistem.....	61
3.5.1 Identifikasi dan Pemodelan Motor DC.....	62
3.5.2 Identifikasi dan Pemodelan ICE.....	74
3.6 Pengujian dan Validasi.....	80
3.7 Perancangan Kontroler PID-LQR .....	82
3.7.1 Desain Simulink PID-LQR .....	85
BAB 4 PENGUJIAN DAN ANALISIS .....	87
4.1 Gambaran Umum Pengujian Sistem .....	87
4.2 Simulasi Sistem.....	87
4.2.1 Simulasi ICE Beban Nominal Tanpa Bantuan Motor Listrik.....	87
4.2.2 Simulasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID- <i>Linear Quadratic Regulator</i> .....	91
4.3 Implementasi Sistem .....	100
4.3.1 Implementasi ICE Beban Nominal Tanpa Bantuan Motor Listrik.....	100
4.3.2 Implementasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID- <i>Linear Quadratic Regulator</i> .....	102
BAB 5 PENUTUP .....	103
5.1 Kesimpulan .....	103
5.2 Saran .....	103
DAFTAR PUSTAKA .....	105
LAMPIRAN .....	107
RIWAYAT HIDUP .....	117

## DAFTAR TABEL

<b>Tabel 2.1</b>	Perbandingan Kapasitas Konfigurasi HEV .....	8
<b>Tabel 2.2</b>	Spesifikasi <i>Board</i> Arduino Uno .....	24
<b>Tabel 2.3</b>	Taksiran Orde ke- $n$ dan Nilai Titik Singgung .....	26
<b>Tabel 2.4</b>	Nilai Konstan Perbandingan $T$ .....	27
<b>Tabel 2.5</b>	Pengaruh Parameter $K_p$ , $K_i$ , dan $K_d$ .....	36
<b>Tabel 3.1</b>	Spesifikasi Mesin Pemotong Rumput 2 Tak .....	44
<b>Tabel 3.2</b>	Spesifikasi Motor DC .....	45
<b>Tabel 3.3</b>	Data Hasil Pembacaan terhadap Sensor <i>Rotary Encoder</i> dan <i>Tachometer Digital</i> .....	60
<b>Tabel 3.4</b>	Perbandingan Data Kecepatan Hasil Linearisasi dengan Kecepatan Asli (dalam rpm) .....	60
<b>Tabel 3.5</b>	Data Percobaan Konstanta Balik Motor DC .....	63
<b>Tabel 3.6</b>	Data Percobaan Kondisi Maksimal dan Nominal .....	64
<b>Tabel 3.7</b>	Data Hasil Pengukuran Parameter .....	66
<b>Tabel 3.8</b>	Hasil pengukuran parameter $L_a$ $R_a$ $L_f$ $R_f$ .....	66
<b>Tabel 3.9</b>	Data Hasil Pengukuran Arus Rem Elektromagnetik .....	67
<b>Tabel 3.10</b>	Persamaan Model dan Validasi <i>Plant</i> ICE Beban Nominal .....	81
<b>Tabel 3.11</b>	Pengujian Metode Pendekatan Model .....	82
<b>Tabel 3.12</b>	Nilai Matriks $\mathbf{P}$ Melalui Penyelesaian ARE .....	84
<b>Tabel 3.13</b>	Nilai Parameter $K_p$ , $K_i$ , dan $K_d$ .....	84
<b>Tabel 4.1</b>	Nilai IAE Saat Arus Beban 0,6 A .....	92
<b>Tabel 4.2</b>	Nilai IAE Saat Arus Beban 0,7 A .....	92
<b>Tabel 4.3</b>	Nilai IAE Saat Beban 0.8 A .....	93





## DAFTAR GAMBAR

<b>Gambar 2.1</b>	HEV Konfigurasi Paralel .....	6
<b>Gambar 2.2</b>	HEV Konfigurasi Seri .....	7
<b>Gambar 2.3</b>	HEV Konfigurasi Seri-Paralel .....	7
<b>Gambar 2.4</b>	<i>Planetary Gear</i> .....	8
<b>Gambar 2.5</b>	Komponen ICE .....	9
<b>Gambar 2.6</b>	Skema Langkah Kerja Mesin 2 Tak .....	10
<b>Gambar 2.7</b>	Kaidah Tangan Kiri .....	12
<b>Gambar 2.8</b>	Gaya Lorentz pada Motor DC.....	13
<b>Gambar 2.9</b>	Motor DC Sederhana .....	15
<b>Gambar 2.10</b>	Karakteristik Motor DC dan Konfigurasinya .....	16
<b>Gambar 2.11</b>	Bentuk Fisik Rem Elektromagnetik .....	17
<b>Gambar 2.12</b>	Ilustrasi Rem Magnetik Arus <i>Eddy</i> .....	18
<b>Gambar 2.13</b>	Konstruksi <i>Timing Belt</i> .....	19
<b>Gambar 2.14</b>	Konfigurasi <i>Gear</i> Satu Poros .....	19
<b>Gambar 2.15</b>	(a) <i>Pitch Circle</i> , (b) <i>Gear</i> dengan 20 Gigi, <i>Pitch Diameter</i> 1'', dan <i>Diametral Pitch</i> 20 .....	20
<b>Gambar 2.16</b>	Konfigurasi <i>Gear</i> Beda Poros .....	21
<b>Gambar 2.17</b>	Konstruksi <i>Rotary Encoder</i> .....	22
<b>Gambar 2.18</b>	<i>Board</i> Arduino Uno .....	23
<b>Gambar 2.19</b>	Respon <i>Step</i> Sistem dengan waktu $t_1$ , $t_2$ , $t_b$ , $T_U$ , dan $T_N$ .....	25
<b>Gambar 2.20</b>	Konfigurasi Fisik Motor DC Konfigurasi Paralel .....	28
<b>Gambar 2.21</b>	Blok Diagram Fisik Motor DC Konfigurasi Paralel .....	29
<b>Gambar 2.22</b>	Diagram Blok Kontroler PID.....	36
<b>Gambar 2.23</b>	Diagram Blok PID-LQR .....	38
<b>Gambar 3.1</b>	Simulator HEV dengan Konfigurasi Paralel.....	41
<b>Gambar 3.2</b>	Diagram Blok Sistem .....	42
<b>Gambar 3.3</b>	Konfigurasi Perangkat Keras Simulator PHEV .....	43
<b>Gambar 3.4</b>	Bentuk Fisik Mesin Pemotong Rumput 2 Tak pada Simulator HEV .....	45
<b>Gambar 3.5</b>	Bentuk Fisik Motor DC pada Simulator HEV .....	46
<b>Gambar 3.6</b>	Transmisi <i>Gear</i> .....	47
<b>Gambar 3.7</b>	Ilustrasi Sensor Posisi pada Pedal Gas.....	49
<b>Gambar 3.8</b>	Potensiometer pada Simulator HEV .....	49
<b>Gambar 3.9</b>	Konfigurasi Rem Elektromagnetik .....	50
<b>Gambar 3.10</b>	Bentuk Fisik Rem Elektromagnetik.....	50
<b>Gambar 3.11</b>	Bentuk Fisik Sistem Minimum Arduino Uno R3 .....	51



<b>Gambar 3.12</b>	Rangkaian <i>Driver</i> Motor DC.....	52
<b>Gambar 3.13</b>	Bentuk Fisik <i>Driver</i> Motor DC .....	52
<b>Gambar 3.14</b>	Skema Rangkaian Penyearah Gelombang Penuh .....	53
<b>Gambar 3.15</b>	Bentuk Fisik Rangkaian Penyearah Gelombang Penuh pada Simulator HEV.....	53
<b>Gambar 3.16</b>	<i>Rotary Encoder</i> pada Simulator HEV .....	54
<b>Gambar 3.17</b>	Modul <i>Optocoupler</i> dan <i>Encoder Disk</i> .....	54
<b>Gambar 3.18</b>	Skema Rangkaian <i>Optocoupler</i> .....	55
<b>Gambar 3.19</b>	Tampilan Program Arduino.....	56
<b>Gambar 3.20</b>	Tampilan <i>Software</i> LabView.....	57
<b>Gambar 3.21</b>	Tampilan MATLAB R2013a .....	58
<b>Gambar 3.22</b>	Tampilan Simulink pada MATLAB R2013a .....	58
<b>Gambar 3.23</b>	Bentuk Fisik <i>Tachometer Digital</i> .....	59
<b>Gambar 3.24</b>	Respon Motor DC saat Diberikan Input Sinyal <i>Step</i> .....	69
<b>Gambar 3.25</b>	Diagram Blok Motor DC.....	69
<b>Gambar 3.26</b>	Diagram Blok Fisik Motor DC Konfigurasi Paralel.....	70
<b>Gambar 3.27</b>	Diagram Blok Fisik Motor DC Konfigurasi Paralel Hasil Linierisasi .....	71
<b>Gambar 3.28</b>	Diagram Blok Pemodelan Strejc .....	74
<b>Gambar 3.29</b>	Respon Kecepatan Putar ICE Beban Minimal.....	76
<b>Gambar 3.30</b>	Respon Kecepatan Putar ICE Beban Nominal .....	78
<b>Gambar 3.31</b>	Respon Kecepatan Putar ICE Beban Maksimal .....	80
<b>Gambar 3.32</b>	Hasil Identifikasi dan Pemodelan ICE Beban Nominal..	81
<b>Gambar 3.33</b>	Desain Simulink PID-LQR.....	85
<b>Gambar 4.1</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,6 A .....	88
<b>Gambar 4.2</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,6 A.....	88
<b>Gambar 4.3</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,7 A .....	89
<b>Gambar 4.4</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,7 A.....	89
<b>Gambar 4.5</b>	Respon Kecepatan Putar ICE dengan Arus Beban 0,8 A .....	90
<b>Gambar 4.6</b>	Sinyal Kesalahan ICE dengan Arus Beban 0,8 A.....	90
<b>Gambar 4.7</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,6 A .....	94
<b>Gambar 4.8</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,6 A....	95
<b>Gambar 4.9</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,6 A .....	95
<b>Gambar 4.10</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,7 A .....	96

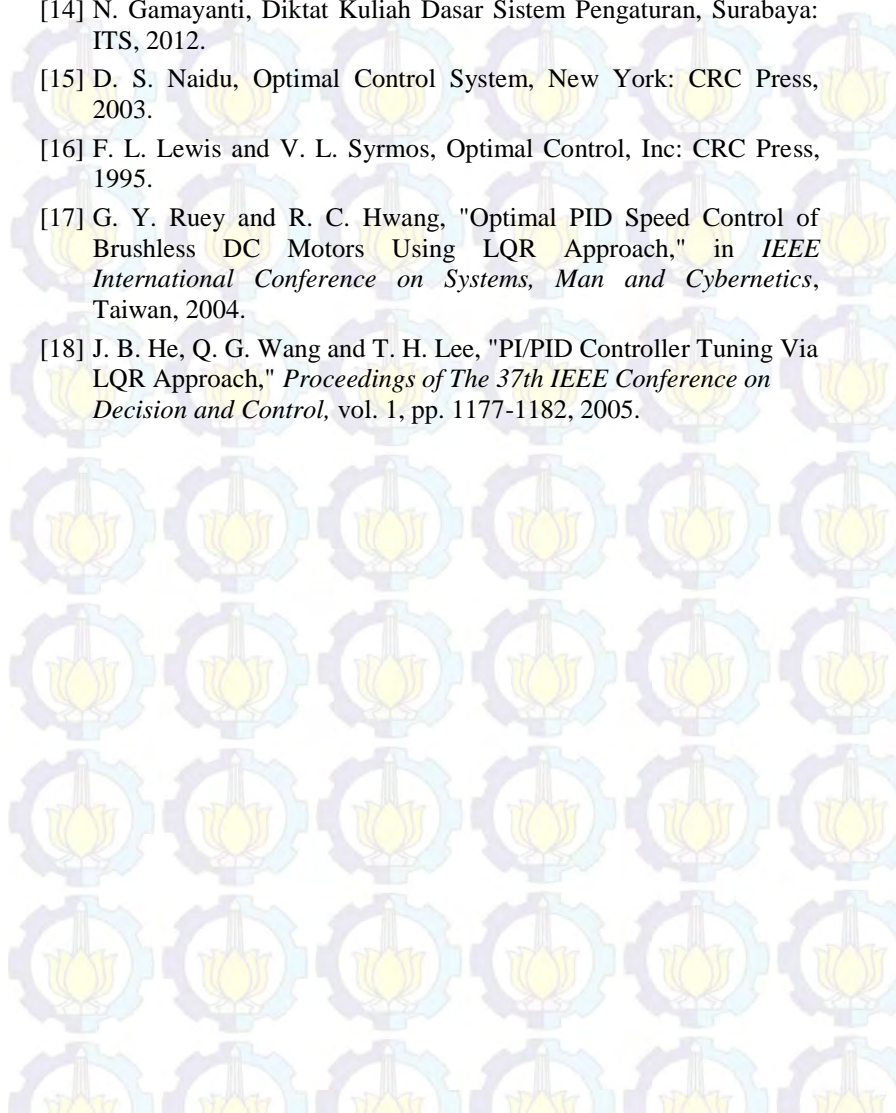
<b>Gambar 4.11</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,7 A ...	97
<b>Gambar 4.12</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,7 A.....	97
<b>Gambar 4.13</b>	Respon Kecepatan Putar HEV dengan Kontroler PID- LQR Saat Arus Beban 0,8 A.....	98
<b>Gambar 4.14</b>	Sinyal Kesalahan dari Sistem Saat Arus Beban 0,8 A ...	99
<b>Gambar 4.15</b>	Sinyal Kontrol Motor DC Saat Arus Beban 0,8 A.....	99
<b>Gambar 4.16</b>	Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 50 VDC.....	101
<b>Gambar 4.17</b>	Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 57,15 VDC.....	101
<b>Gambar 4.18</b>	Respon Kecepatan Putar HEV Hasil Implementasi Menggunakan Kontroler PID-LQR dengan 3 Variasi Beban .....	102





## DAFTAR PUSTAKA

- [1] M. Mohebbi, M. Charkhgard and M. Farrokhi, "Optimal Neuro-Fuzzy Control of Parallel Hybrid Electric Vehicles", *Vehicle Power and Propulsion, IEEE Conference*, pp.26-30, 7-9 Sept, 2005
- [2] W. A. Candra, Pengaturan Kecepatan pada Parallel Hybrid Electric Vehicle Menggunakan Linear Quadratic Regulator Berdasarkan Particle Swarm Optimization, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [3] G. Satrio Aji Wibowo, Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle, Surabaya: Jurusan Teknik Elektro ITS, 2010.
- [4] W. Sriwidodo, Perancangan dan Implementasi Pengendalian Traksi Berbasis Neuro Fuzzy untuk Simulator Kendaraan Hybrid pada Mode Akselerasi, Surabaya: Jurusan Teknik Elektro ITS, 2010.
- [5] Y. Du, J. Gao, L. Yu, J. Song, F. Zhao and W. Zhan, "HEV System Based on Electric Variable Transmission," in *Vehicle Power and Propulsion Conference IEEE*, 2009.
- [6] A. Thonthowi, Pengaturan Kecepatan pada Parallel Hybrid Electric Vehicle Menggunakan Self Tuning Proportional Integral Derivative Berdasarkan Genetic Algorithm, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [7] A. Rahadi, "Cara Kerja Mesin 2 Tak dan 4 Tak," Juli 2014.
- [8] A. Endarwati, Perancangan Sistem Pengaturan Kecepatan pada Simulator Parallel Hybrid Electric Vehicle (PHEV) Menggunakan Metode State Dependent-Linear Quadratic Regulator, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [9] D. SDP, "Drive Your System With Timing Belt," Juli 2013.
- [10] R. Hardiningrat, Pembuatan Alat Ukur Kedalaman Lubang Sumur Bor, Bandung: Teknik Mesin Universitas Pasundan, 2013.
- [11] "Index of Arduino Board," November 2013. [Online]. Available: <http://www.arduino.cc/en/Main/arduinoBoardUno>. [Accessed Mei 2015].
- [12] I. Pavel Jakoubek, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods," in *Konference Studenske Tvuurci-Cinnosti*, 2009.

- 
- [13] K. Ogata, Modern Control Engineering, Prentice-Hall Inc, 1970.
- [14] N. Gamayanti, Diktat Kuliah Dasar Sistem Pengaturan, Surabaya: ITS, 2012.
- [15] D. S. Naidu, Optimal Control System, New York: CRC Press, 2003.
- [16] F. L. Lewis and V. L. Syrmos, Optimal Control, Inc: CRC Press, 1995.
- [17] G. Y. Ruey and R. C. Hwang, "Optimal PID Speed Control of Brushless DC Motors Using LQR Approach," in *IEEE International Conference on Systems, Man and Cybernetics*, Taiwan, 2004.
- [18] J. B. He, Q. G. Wang and T. H. Lee, "PI/PID Controller Tuning Via LQR Approach," *Proceedings of The 37th IEEE Conference on Decision and Control*, vol. 1, pp. 1177-1182, 2005.

## RIWAYAT HIDUP



Penulis bernama lengkap Fanniesha Hamada, Lahir di kota Jakarta pada tanggal 13 Januari 1994 dan merupakan anak kedua dari 3 bersaudara dari pasangan Arifin dan Aryani Nilasari. Setelah lulus dari SMAN 2 Kota Tangerang Selatan pada tahun 2011, penulis melanjutkan studinya di Institut Teknologi Sepuluh Nopember Surabaya (ITS), Jurusan Teknik Elektro dan mengambil bidang studi Teknk Sistem Pengaturan. Pada tahun kedua dan ketiga perkuliahan, penulis aktif dalam bidang organisasi dan kepanitiaan *event* di lingkungan kampus maupun di luar kampus. Pada bulan Juni 2015, penulis mengikuti seminar dan ujian Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro.



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jumlah kendaraan bermotor di Indonesia terus mengalami peningkatan. Hal ini menunjukkan betapa besar kontribusi pencemaran udara dari kendaraan transportasi selain dari kegiatan industri dan rumah tangga. Dari sekian banyak penggunaan kendaraan bermotor tersebut maka akan berbanding lurus dengan penggunaan bahan bakar minyak. Penggunaan bahan bakar minyak untuk kendaraan bermotor yang semakin meningkat menyebabkan menipisnya cadangan energi minyak bumi.

Seiring berkembangnya isu efek rumah kaca karena pencemaran udara dan krisis energi, maka pengembang teknologi terus melakukan inovasi. Sektor transportasi juga melakukan inovasi dibidang teknologi kendaraan. *Hybrid Electric Vehicle* (HEV) merupakan suatu kendaraan dengan konsep ramah lingkungan dan hemat energi yang diharapkan menjadi salah satu alternatif menanggulangi efek rumah kaca dan krisis energi.

Kendaraan *Hybrid Electric Vehicle* (HEV) dengan mesin *hybrid* adalah kendaraan yang menggunakan dua atau lebih tenaga sebagai sumber penggerak [1], yang merupakan gabungan antara mesin bakar dan motor listrik. Pada mesin *hybrid*, tenaga yang digunakan untuk menggerakkan kendaraan bisa berasal dari ICE, motor listrik, maupun gabungan diantara keduanya [1]. Kelebihan dari mesin *hybrid* adalah saat ICE tidak mampu mempertahankan kecepatan maka motor listrik akan membantu memberikan tenaga [2]. Ketidakmampuan dalam mempertahankan kecepatan tersebut biasanya muncul saat terjadi pembebanan lebih pada kendaraan.

Metode *PID-Linear Quadratic Regulator* (PID-LQR) digunakan untuk mengatur kecepatan motor listrik, sehingga dapat membantu ICE saat terjadi pembebanan lebih yang dapat menurunkan kecepatan pada kendaraan. Metode PID-LQR merupakan metode PID optimal dengan pendekatan LQR untuk menentukan parameter  $K_p$ ,  $K_i$  dan  $K_d$  pada kontroler PID. Dengan menggunakan metode ini diharapkan terjadinya penurunan kecepatan akibat efek pembebanan berlebih dapat diatasi dengan adanya pembagian kerja antara dua tenaga sebagai sumber penggerak. Pada Tugas Akhir ini digunakan Simulator *Parallel Hybrid*

*Electric Vehicle* (PHEV). Simulator PHEV memadukan dua buah pembangkit torsi untuk menanggung beban yang terjadi, dengan ICE bergerak sebagai penggerak utama dan motor DC bertindak sebagai penggerak pembantu [3]. Simulator PHEV yang digunakan terdiri dari mesin pemotong rumput sebagai ICE, motor DC paralel sebagai motor listrik, dan rem elektromagnetik sebagai beban pengereman yang telah dirancang sebelumnya pada Tugas Akhir tahun 2010 [4].

## 1.2 Perumusan Masalah

Saat ICE mendapat beban dari rem elektromagnetik maka kecepatan putar ICE akan menurun sehingga kecepatan HEV tidak sesuai dengan *output* yang diharapkan serta mempengaruhi kinerja dari sistem. Efek pembebanan tersebut biasanya terjadi ketika kendaraan berada pada tanjakan. Permasalahan yang menjadi topik dalam Tugas Akhir ini adalah bagaimana mengatur kecepatan motor listrik agar dapat membantu ICE untuk menanggung beban yang berlebih sehingga dapat mencapai kecepatan putar yang sesuai. Oleh karena itu, dibutuhkan suatu kontroler untuk melakukan pengaturan kerja dari motor listrik agar bekerja sesuai dengan kebutuhan yang diinginkan.

## 1.3 Batasan Masalah

Batasan masalah dalam penyelesaian Tugas Akhir ini adalah sebagai berikut :

1. *Plant* yang digunakan adalah simulator HEV yang terdiri dari ICE sebagai penggerak utama yang dioperasikan secara manual tanpa diberikan aksi kontrol, motor listrik yaitu motor DC sebagai penggerak pembantu dan rem elektromagnetik sebagai pemberi efek pembebanan.
2. Sumber tegangan untuk mengoperasikan motor listrik berasal dari jala-jala PLN sehingga tidak ada mekanisme *charging*, *discharging* dan *regenerative braking*.
3. Pengaturan regulasi pada simulator HEV dilakukan oleh motor DC.
4. Motor listrik dioperasikan dengan *range* sumber tegangan 0 sampai 150 V.
5. Model nominal digunakan untuk mengatasi pembebanan pada *range* pembebanan nominal.



## 1.4 Tujuan

Tujuan dari pelaksanaan Tugas Akhir ini adalah merancang kontroler *PID-Linear Quadratic Regulator* (PID-LQR) untuk pengaturan kecepatan pada simulator PHEV sehingga dapat mengatasi permasalahan adanya efek pembebanan berlebih pada HEV. Hasil dari perancangan kontroler diharapkan dapat memperoleh performansi terbaik untuk pengaturan kecepatan pada pembebanan yang berbeda-beda sehingga kecepatan yang diinginkan dapat terus dipertahankan.

## 1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima BAB dengan sistematika penulisan sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang, perumusan masalah, tujuan, sistematika penulisan dan relevansi dari Tugas Akhir ini.

### **BAB II DASAR TEORI**

Bab ini berisikan penjelasan mengenai konsep dasar yang berkaitan dengan pengerjaan Tugas Akhir ini.

### **BAB III PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan perangkat keras dan perangkat lunak, pengujian sistem, identifikasi dan pemodelan *plant*, serta perancangan kontroler PID-LQR.

### **BAB IV PENGUJIAN DAN ANALISA**

Bab ini memuat hasil simulasi dan implementasi kontroler PID-LQR dengan kondisi beban yang berubah-ubah dan menganalisa dari data yang telah didapatkan.

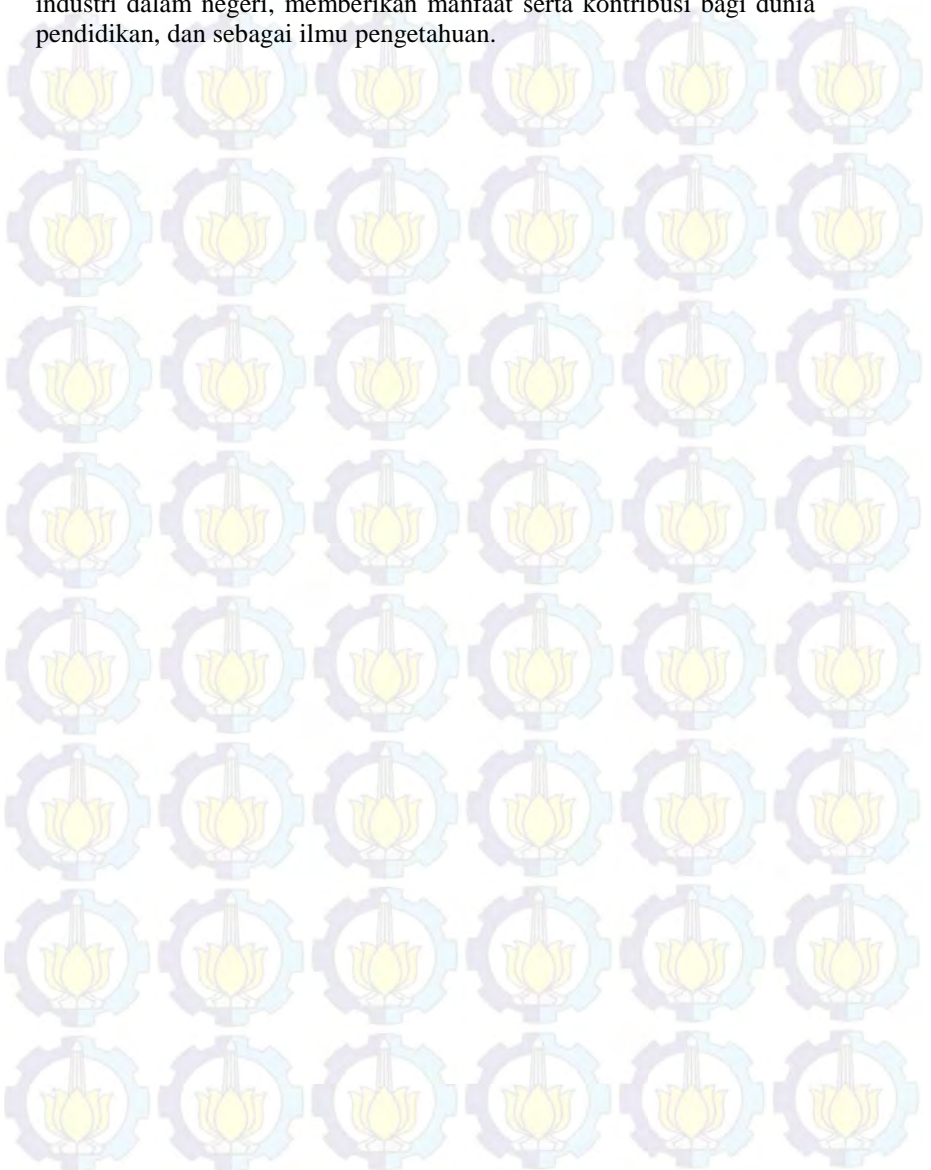
### **BAB V PENUTUP**

Berisi pemaparan kesimpulan dan saran dari hasil pembahasan yang telah diperoleh serta sebagai pertimbangan bagi pembaca untuk pengembangan dari Tugas Akhir ini selanjutnya.

## 1.6 Relevansi

Hasil yang diperoleh dari penelitian Tugas Akhir ini diharapkan dapat memberikan manfaat tentang penelitian dalam bidang pengaturan kendaraan *hybrid* baik dalam referensi kontroler pada pengaturan

kecepatan untuk mendapatkan performansi regulasi yang lebih baik. Selain itu, juga membantu pengembangan teknologi otomotif dan industri dalam negeri, memberikan manfaat serta kontribusi bagi dunia pendidikan, dan sebagai ilmu pengetahuan.



## **BAB 2**

### **DASAR TEORI**

#### **2.1 *Hybrid Electric Vehicle (HEV)* [5]**

Kendaraan listrik hibrid berkembang dengan pesat di era 1990-an. Diinisiasi oleh Toyota dengan produk kenamaannya, Toyota Prius, dan kini telah banyak perusahaan mobil yang memproduksi kendaraan listrik hibrid. Kendaraan jenis ini menjadi sangat populer karena kemampuannya menghemat penggunaan bahan bakar yang saat ini masih didominasi oleh bahan bakar fosil. Selain itu, bumi dengan perubahan suhu yang semakin ekstrim mendesak manusia untuk berusaha lebih keras lagi mengurangi emisi CO<sup>2</sup>.

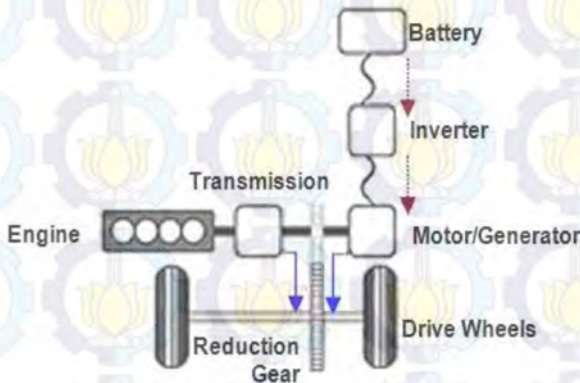
Pada dasarnya, kendaraan listrik hibrid bekerja dengan mengkombinasikan pemanfaatan mesin bakar dan motor listrik untuk mencapai penggunaan bahan bakar yang lebih ekonomis. Pemanfaatan motor listrik diharapkan dapat membantu mesin bakar bekerja dengan temperatur mesin yang tidak terlalu panas. Hal ini dicapai dengan menjaga mesin bakar bekerja dengan kecepatan yang relatif konstan. Selain itu, sistem pengereman regeneratif pada kendaraan listrik hibrid, dapat memberikan suplai listrik ke baterai sehingga energi pengereman tidak terbuang sia-sia sebagai panas seperti pada kendaraan konvensional. Bantuan dari motor listrik diberikan saat kendaraan menanjak, menurun, percepatan, mulai berjalan dan perlambatan. Saat kendaraan mulai berjalan, mesin bakar dinyalakan. Pada kecepatan mencapai 1000 rpm, motor listrik dimatikan dan mesin bakar digunakan. Saat kendaraan berada di turunan, mesin bakar mengisi baterai jika baterai dalam tidak terisi penuh. Saat percepatan atau di tanjakan, motor listrik dan mesin bakar menyuplai tenaga gerak secara bersamaan (pada konfigurasi seri: sumber energi motor listrik didapat dari baterai dan mesin bakar). Saat perlambatan, pengereman regeneratif mengisi baterai.

##### **2.1.1 HEV Konfigurasi Paralel [3] [6]**

Pada konfigurasi paralel, terdapat dua jalur aliran daya, aliran mekanik dan aliran elektrik. Setiap aliran daya bisa menjalankan kendaraan secara individu maupun bersama. Motor listrik bisa difungsikan sebagai motor maupun generator. Pengaturan kerja motor listrik sangat penting sehingga tidak memboroskan energi yang tersimpan di dalam baterai. Kelemahan konfigurasi ini adalah kapasitas



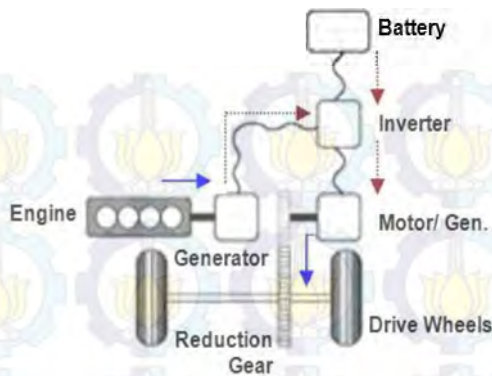
*charge* baterai tidak terlalu besar dan bersifat diskontinyu serta kesulitannya menentukan kondisi kerja mesin bakar yang optimal karena langsung terhubung ke sistem transmisi. Namun di sisi lain, tenaga gerak yang dirasakan dapat setara dengan kendaraan konvensional. Konfigurasi paralel lebih cocok diterapkan pada kendaraan yang jarak tempuhnya jauh (fluktuasi kecepatannya relatif rendah) atau yang membutuhkan daya yang besar (efisiensi meningkat seiring dengan peningkatan daya). HEV konfigurasi paralel dapat dilihat pada **Gambar 2.1**.



**Gambar 2.1** HEV Konfigurasi Paralel [6]

### 2.1.2 HEV Konfigurasi Seri [3] [6]

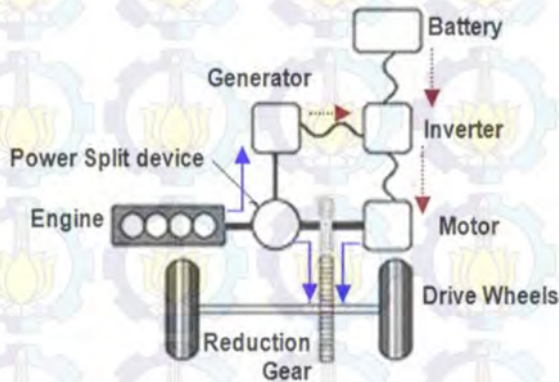
Konfigurasi seri seperti yang ditunjukkan pada **Gambar 2.2**, hanya menggunakan motor listrik sebagai penggerak yang terhubung langsung dengan sistem transmisi. Tenaga gerak dari mesin bakar selalu diubah ke dalam listrik oleh generator listrik sehingga efisiensi sistem sangat bergantung pada efisiensi generator dan motor listrik yang digunakan. Daya listrik yang dihasilkan generator dapat menyuplai motor listrik, jika dibutuhkan. Jika tidak, akan digunakan untuk mengisi baterai. Karena operasi mesin bakar tidak tergantung dari kecepatan kendaraan dan beban, maka mesin bakar bisa beroperasi di kondisi optimal sepanjang waktu. Kekurangan dari konfigurasi ini adalah respon yang tidak terlalu cepat bila diinginkan akselerasi tiba-tiba. Konfigurasi seri sangat cocok untuk penggunaan kendaraan dengan frekuensi jalan dan berhenti yang tinggi seperti pada bis-bis di perkotaan.



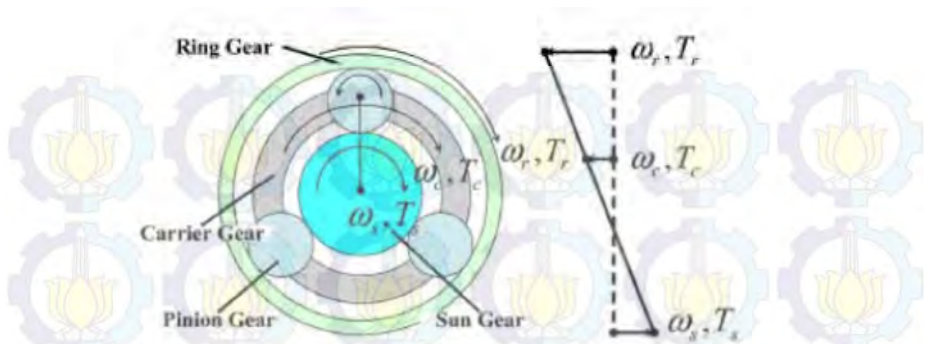
**Gambar 2.2** HEV Konfigurasi Seri [6]

### 2.1.3 HEV Konfigurasi Seri-Paralel [3] [6]

HEV dengan konfigurasi seri – paralel menggunakan *power split device* seperti yang ditunjukkan pada **Gambar 2.3**, sehingga dapat disebut konfigurasi *power split*. Dengan adanya *power split device* kekurangan pada konfigurasi paralel dan seri dapat teratasi. Sistem ini dikembangkan oleh Toyota pertama kali pada Toyota Prius generasi pertama. Prinsip dasar dari *power split device* ini adalah menggunakan sistem *planetary gear* seperti pada **Gambar 2.4**.



**Gambar 2.3** HEV Konfigurasi Seri-Paralel [6]



**Gambar 2.4** Planetary Gear [3]

Planetary gear terdiri dari tiga komponen, antara lain: *sun gear* yang berada di pusat, *ring gear* yang berada di posisi paling luar, *pinion gear* yang menghubungkan *sun gear* dan *ring gear*, dan *carrier gear* yang menghubungkan semua *pinion gear*.

Dari **Gambar 2.4** dapat diamati bahwa *sun gear* terhubung dengan generator, *planetary carrier* terhubung dengan motor bakar, dan *ring gear* terhubung dengan motor listrik atau *shaft* utama. Desain yang demikian, memungkinkan pengisian baterai secara kontinu dari ICE sekalipun sedang digunakan untuk menggerakkan kendaraan. Jadi apabila tiba-tiba dibutuhkan akselerasi, maka respon kendaraan akan cepat dan kebutuhan daya motor listrik terus terjaga.

Berikut adalah perbandingan kapasitas konfigurasi paralel, seri, dan *power split device* (seri-paralel) dapat dilihat pada **Tabel 2.1**.

**Tabel 2.1** Perbandingan Kapasitas Konfigurasi HEV [3]

Type	Efisiensi bahan bakar			Performansi	
	<i>Idling Stop</i>	Recovery Energy	Efisiensi	Akselerasi	Kestabilan kecepatan
Seri	●	◎	●	●	◎
Paralel	●	●	●	●	◎
Seri-Paralel	◎	◎	◎	●	●



Sangat baik



Baik



Tidak Baik

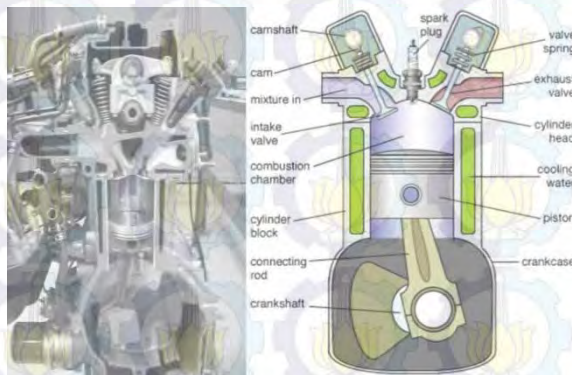


## 2.2 Simulator Hybrid Electric Vehicle

Pada penelitian ini digunakan simulator *Hybrid Electric Vehicle* dengan konfigurasi paralel yang terdiri dari ICE 2 tak sebagai penggerak utama, motor DC sebagai penggerak pendukung, dan beban berupa rem elektromagnetik.

### 2.2.1 Internal Combustion Engine (ICE) 2 Tak [7]

ICE merupakan mesin pembakaran dalam dimana langkah kerjanya beroperasi berdasar prinsip satu siklus yang bekerja secara terus menerus. ICE dibagi berdasarkan jumlah *stroke* atau kayuhan (tak). Jumlah *stroke* menentukan jumlah proses yang dilakukan dalam satu siklus. Komponen ICE dapat dilihat pada **Gambar 2.5**.



**Gambar 2.5** Komponen ICE [7]

Menurut langkah kerjanya ICE dibagi menjadi mesin dengan proses 2 langkah (tak) dan mesin dengan proses 4 langkah (tak). Mesin 2 tak adalah mesin pembakaran dalam yang dalam satu siklus pembakaran terjadi dua langkah piston, berbeda dengan mesin 4 tak yang mempunyai empat langkah piston dalam satu siklus pembakaran, meskipun keempat proses (*intake*, kompresi, tenaga, pembuangan) juga terjadi. ICE 2 tak menyelesaikan keempat tahap siklus, yaitu *intake stroke*, *compression stroke*, *power stroke*, dan *exhaust stroke* hanya dalam dua langkah (kayuhan).

Untuk memahami prinsip kerjanya, perlu dimengerti istilah baku yang berlaku dalam teknik otomotif :

- a. TMA (Titik Mati Atas) atau TDC (*Top Dead Centre*), posisi piston berada pada titik paling atas dalam silinder mesin atau piston berada pada titik paling jauh dari poros engkol (*crankshaft*).
- b. TMB (Titik Mati Bawah) atau BDC (*Bottom Dead Centre*), posisi piston berada pada titik paling bawah dalam silinder mesin atau piston berada pada titik paling dekat dengan poros engkol (*crankshaft*).
- c. Ruang bilas yaitu ruangan dibawah piston dimana terdapat poros engkol (*crankshaft*), sering disebut dengan bak engkol (*crankcase*) berfungsi gas hasil campuran udara, bahan bakar dan pelumas bisa tercampur lebih merata.
- d. Pembilasan (*scavenging*) yaitu proses pengeluaran gas hasil pembakaran dan proses pemasukan gas untuk pembakaran dalam ruang bakar.

## MESIN 2 LANGKAH



**Gambar 2.6** Skema Langkah Kerja Mesin 2 Tak [7]

Prinsip kerja dari ICE dengan siklus 2 langkah atau mesin 2 tak seperti yang ditunjukkan pada **Gambar 2.6** dapat dijelaskan sebagai berikut :

- a. **Langkah ke-1 (Isap dan Kompresi) :**
  - a. Piston bergerak dari TMA ke TMB.
  - b. Saat bergerak dari TMA ke TMB, piston akan menekan ruang bilas yang berada di bawahnya. Semakin jauh piston



meninggalkan TMA menuju TMB akan semakin meningkat pula tekanan di ruang bilas.

- c. Pada titik tertentu, piston (*ring piston*) akan melewati lubang pembuangan gas dan lubang pemasukan gas. Posisi masing-masing lubang tergantung dari desain perancang. Umumnya *ring piston* akan melewati lubang pembuangan terlebih dahulu.
- d. Pada saat *ring piston* melewati lubang pembuangan, gas di dalam ruang bakar keluar melalui lubang pembuangan.
- e. Pada saat *ring piston* melewati lubang pemasukan, gas yang tertekan di dalam ruang bilas akan terpompa masuk ke dalam ruang bakar, sekaligus mendorong keluar gas yang ada di dalam ruang bakar menuju lubang pembuangan.
- f. Piston terus menekan ruang bilas sampai titik TMB, sekaligus memompa gas dalam ruang bilas menuju ke dalam ruang bakar.

**b. Langkah ke-2 (Pembakaran dan Buang) :**

- a. Piston bergerak dari TMB ke TMA.
- b. Saat bergerak dari TMB ke TMA, piston akan menghisap gas hasil percampuran udara, bahan bakar dan pelumas ke dalam ruang bilas. Percampuran ini dilakukan oleh karburator atau sistem injeksi.
- c. Saat melewati lubang pemasukan dan lubang pembuangan, piston akan mengkompresi gas yang terjebak di dalam ruang bakar.
- d. Piston akan terus mengkompresi gas dalam ruang bakar sampai TMA.
- e. Beberapa saat sebelum piston sampai di TMA, busi akan menyala untuk membakar gas dalam ruang bakar. Waktu nyala busi tidak terjadi saat piston sampai ke TMA, melainkan terjadi sebelumnya. Ini dimaksudkan agar puncak tekanan akibat pembakaran dalam ruang bakar bisa terjadi saat piston mulai bergerak dari TMA ke TMB, karena proses pembakaran membutuhkan waktu untuk bisa membuat gas terbakar dengan sempurna oleh nyala api busi.

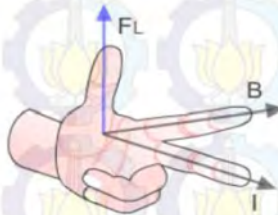
Karena mesin 2 tak dalam 1 putaran kruk as/*crankshaft* melaksanakan 4 siklus, mesin 2 tak ini lebih responsif dan akselerasinya bagus. Akan tetapi, mesin ini mengeluarkan tenaga yang besar pada saat putaran/rpm tinggi sehingga membuat mesin ini membutuhkan bahan bakar yang relatif banyak. Akan tetapi, mesin ini menghasilkan tenaga yang lebih besar dibandingkan mesin 4 tak. Bahan bakar mesin ini tak

hanya bensin, tetapi dapat dioplos dengan oli khusus yang biasa disebut oli samping untuk melumasi bagian dalam mesin sehingga oli mesin hanya melumasi bagian transmisi. Hal ini menyebabkan mesin 2 tak mengeluarkan asap karena membakar oli samping.

### 2.2.2 Motor Listrik DC [2] [8]

Mesin listrik adalah alat yang dapat mengubah energi mekanik menjadi energi listrik atau dari energi listrik menjadi energi mekanik. Ketika mesin listrik mengubah energi mekanik menjadi energi listrik menggunakan alat konversi yang disebut generator. Sebaliknya dengan alat konversi yang disebut motor, energi listrik dapat diubah menjadi energi mekanik. Motor listrik bekerja dengan memanfaatkan gaya-gaya magnet. Motor listrik dibagi menjadi dua jenis, yaitu motor arus bolak-balik (AC) dan motor arus searah (DC).

Prinsip kerja motor DC berawal dari prinsip bahwa apabila terdapat konduktor yang menghantarkan arus dan diletakkan pada medan magnet maka akan menghasilkan suatu gaya. Arah gaya yang dihasilkan ditentukan melalui kaidah tangan kiri seperti dapat dilihat pada **Gambar 2.7**, dengan ibu jari diibaratkan sebagai arah gaya, jari telunjuk sebagai arah medan magnet, dan jari tengah diibaratkan sebagai arah arus yang mengalir pada konduktor.



**Gambar 2.7** Kaidah Tangan Kiri [8]

Medan magnet menghasilkan magnet permanen yang nilainya konstan, sedangkan komutator dan sikat berfungsi menyalurkan arus listrik dari sumber di luar motor ke dalam kumparan jangkar.

Medan *stator* menghasilkan  $\phi$  dari kutub *U* ke kutub *S*. Sikat arang menyentuh terminal kumparan *rotor* di bawah kutub. Bila sikat arang dihubungkan pada satu sumber arus searah di luar tegangan *V*, maka satu arus *I* masuk ke terminal kumparan *rotor* di bawah kutub *U* dan keluar dari terminal di bawah kutub *S*. Dengan adanya fluks *stator*

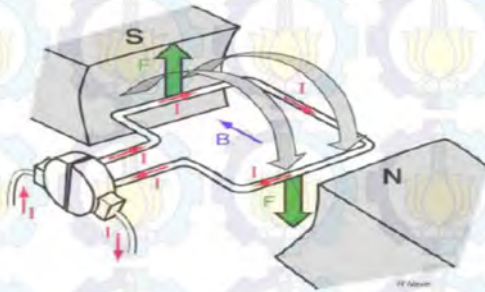
dan arus *rotor* akan menghasilkan satu gaya  $F$  bekerja pada kumparan yang dikenal dengan gaya Lorentz. Arah  $F$  yang ditunjukkan pada **Gambar 2.8** menghasilkan torsi yang memutar *rotor* ke arah yang berlawanan dengan jarum jam. Besar gaya Lorentz yang dihasilkan dapat ditentukan berdasarkan **Persamaan (2.1)**.

$$F = BIL\sin\theta \quad (2.1)$$

dengan,

- $F$  : Gaya Lorentz (N)
- $B$  : Kerapatan fluks magnet (Weber/m )
- $I$  : Arus yang mengalir (A)
- $L$  : Panjang kawat konduktor (m)
- $\theta$  : Sudut yang terbentuk antara arah arus medan dengan arah arus yang mengalir pada kawat (Derajat)

Saat arus mengalir melewati kawat yang memotong medan magnet seperti pada **Gambar 2.8**, maka akan timbul gaya Lorentz. Gaya Lorentz inilah yang menimbulkan torsi pada *shaft* motor. Karena motor DC membangkitkan torsi, maka seringkali disebut sebagai pembangkit torsi.



**Gambar 2.8** Gaya Lorentz pada Motor DC [2]

Komutator berfungsi untuk membalik polaritas, sehingga arus mengalir dengan arah yang berlawanan dari sebelumnya. Perhatikan **Gambar 2.8**, saat kawat berputar sesuai arah gaya hingga posisi tegak lurus gaya yang bekerja tetap sama. Namun setelah itu diperlukan gaya yang berlawanan agar kawat tetap berputar. Untuk membalik gaya,



maka arah arus perlu dibalik polaritasnya dari kedua ujung kawat. Komutator akan terbalik dengan sendirinya setelah kawat berputar selama setengah putaran. Komutator secara fisik berupa segmen-segmen, biasanya terbuat dari tembaga dan terletak pada rotor. Sikat arang (*brushes*) terbuat dari karbon, ditekan pada komutator dengan menggunakan pegas. Sikat arang ini membawa arus listrik dari sumber menuju kumparan jangkar. Pada komutator, setiap segmen dipisahkan oleh isolator seperti mika atau *polymer*. Setiap segmen juga dihubungkan dengan beberapa kumparan pada kumparan jangkar.

Struktur fisik dari motor DC terdiri dari dua bagian yaitu *stator* (bagian yang diam) dan *rotor* (bagian yang berputar). Pada bagian *stator* merupakan tempat kumparan medan yang berfungsi untuk menghasilkan fluks magnet sedangkan pada bagian *rotor* terdapat rangkaian jangkar seperti kumparan jangkar (*armature*) yang menghasilkan gaya putar, komutator dan sikat. Kumparan jangkar pada dasarnya adalah kawat yang dialiri arus, sehingga menimbulkan gaya Lorentz. Disebut kumparan, karena jumlah kawatnya yang banyak. Dengan demikian semakin banyak kawat, maka gaya Lorentz yang dihasilkan akan semakin besar dan kontinyu. Torsi yang terbentuk mengikuti **Persamaan (2.2)** berikut:

$$\tau = KI_a \phi \quad (2.2)$$

dengan,

$\tau$  : Torsi yang dibangkitkan (Nm)

$K$  : Konstanta mesin

$I_a$  : Arus jangkar (Ampere)

$\phi$  : Fluks magnet (Weber)

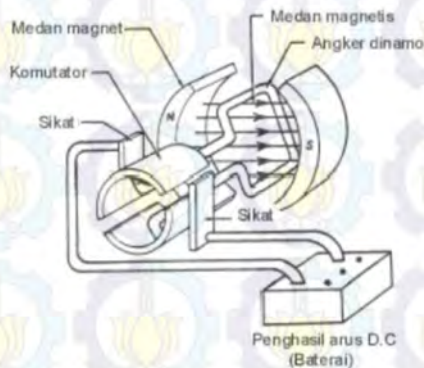
Bila kumparan jangkar dari motor berputar dalam medan magnet dan memotong fluks utama, sesuai dengan hukum induksi elektromagnetik, maka pada kumparan jangkar akan timbul gaya gerak listrik (ggl) induksi yang arahnya sesuai dengan kaidah tangan kanan, di mana arahnya berlawanan dengan tegangan yang diberikan kepada jangkar atau tegangan terminal. Karena arahnya melawan, maka ggl induksi ini disebut ggl lawan dengan **Persamaan (2.3)**.

$$E_a = cn\phi \quad (2.3)$$

dengan,

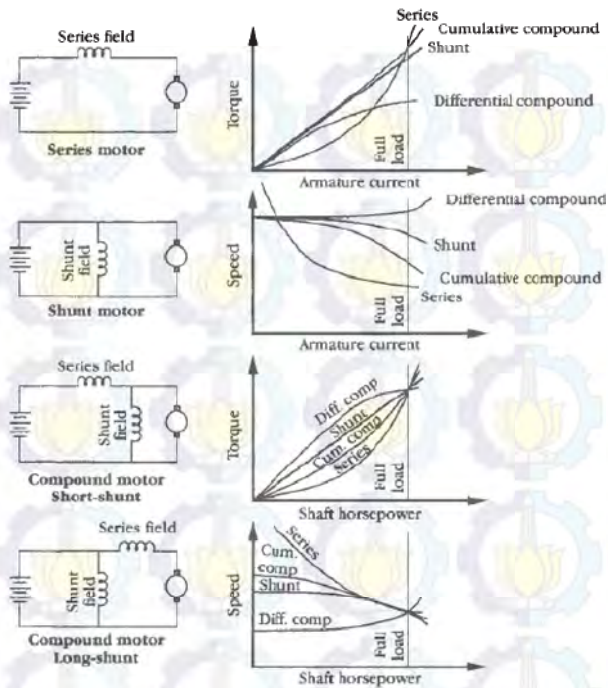
$E_a$  : Gaya gerak listrik induksi (Volt)  
 $n$  : Kecepatan putaran (rpm)  
 $\phi$  : Fluksi setiap kutub (Weber)  
 $c$  : Konstanta

Berdasarkan konstruksi nya, motor DC dapat dibedakan menjadi *brushed*, *brushless* dan *permanent magnet*. Bentuk motor DC yang paling sederhana adalah memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen yang dapat dilihat pada **Gambar 2.9**. Yang sering digunakan adalah motor DC *brushed* karena konstruksinya yang sederhana, sehingga berharga lebih murah dan juga memiliki kemampuan yang cukup baik. Motor DC *brushed* terdiri dari beberapa konfigurasi, yaitu paralel, seri, dan *compound* (campuran) yang masing masing memiliki karakteristik dan konfigurasi yang berbeda-beda (dapat dilihat pada **Gambar 2.10**).



**Gambar 2.9** Motor DC Sederhana [8]

Untuk mengatur kecepatan ataupun torsi pada motor DC, terlebih dahulu harus mengetahui dan mengerti mengenai karakteristik tipe motor DC yang akan digunakan. Pada setiap konfigurasi motor DC, memiliki karakteristik dan penurunan persamaan model matematika yang berbeda pula. Dalam pemilihan penggunaan motor DC, maka harus diperhatikan pula konfigurasi dan karakteristik dari setiap konfigurasi.



**Gambar 2.10** Karakteristik Motor DC dan Konfigurasiya [2]

### 2.2.3 Rem Elektromagnetik [2] [8]

Sistem pengereman adalah sebuah sistem yang berfungsi untuk menghalangi suatu gerakan. Sistem pengereman bertugas mengkonversi energi mekanis (energi gerak) suatu benda ke bentuk lain sehingga gerakan benda tersebut menjadi berkurang. Pada mobil yang sedang bergerak misalnya, sistem rem mengkonversikan energi gerak mobil menjadi panas yang terbuang melalui gesekan pada kanvas rem dengan piringan ataupun tromol roda. Dengan kata lain sistem rem konvensional (sebut saja rem tromol ataupun rem cakram) membuang begitu saja energi panas yang terjadi pada saat proses pengereman.

Terdapat dua tipe rem, yaitu tipe rem mekanik dan elektromagnetik. Tipe rem mekanik menggunakan gaya gesek yang menimbulkan gaya lawan terhadap gaya gerak, tetapi tipe rem ini memiliki kekurangan antara lain bisa menimbulkan panas yang berlebihan sehingga



mengurangi gaya gesek yang terjadi. Tipe rem elektromagnetik memanfaatkan gesekan dua permukaan untuk menghasilkan gaya lawan terhadap gaya gerak, yaitu menggunakan gaya magnet untuk menimbulkan gaya lawan. Rem elektromagnetik disebut juga rem magnetik arus *eddy*. Saat ini penggunaannya sudah meluas karena lebih menghemat biaya pemeliharaan dan penggantian komponen yang bergesekan. Bentuk fisik rem elektromagnetik dapat dilihat pada **Gambar 2.11**.

Prinsip dasar dari rem magnetik arus *eddy* menggunakan hukum Faraday dan hukum Lenz. Pada hukum Faraday dijelaskan bahwa besar ggl induksi atau arus induksi tergantung pada laju perubahan fluks dan banyaknya lilitan. Pada hukum Lenz dijelaskan bahwa arus induksi yang timbul arahnya sedemikian sehingga menimbulkan medan magnet induksi yang melawan arah perubahan medan magnet. Dari penjelasan ini dapat disimpulkan bahwa arus induksi akan timbul jika terdapat perubahan fluks. Perubahan fluks ini akan terjadi apabila konduktor tersebut berputar. Semakin cepat putaran maka akan semakin besar pula laju perubahan fluks sehingga arus induksi yang dihasilkan akan semakin besar, begitu juga sebaliknya. Arus induksi yang dihasilkan tadi akan menimbulkan medan magnet induksi yang berlawanan arah dengan arah perubahan medan magnet sehingga akan menjadikan putaran dari konduktor melambat.



**Gambar 2.11** Bentuk Fisik Rem Elektromagnetik [2]

Komponen dasar rem magnetik arus *eddy* adalah sumber magnet yang diposisikan tetap dan konduktor diamagnetik (tidak memiliki sifat magnet namun memiliki sifat penghantar listrik) yang terhubung dengan

poros. Arus *eddy* yang timbul pada sebuah cakram yang dikelilingi medan magnet dapat dilihat pada **Gambar 2.12**.



**Gambar 2.12** Ilustrasi Rem Magnetik Arus *Eddy* [8]

#### 2.2.4 *Timing Belt* [2] [9]

*Belt* mempunyai fungsi yang dapat dikatakan vital, karena bila putus maka AC, kelistrikan dan mesin bisa *drop* bahkan mati. *Belt* tersebut diharuskan untuk kuat, tahan lama, keras sekaligus elastis. *Belt* sering kali digunakan untuk menghubungkan dua buah roda gigi motor untuk merubah kecepatan ataupun kecepatan sudut yang bertujuan untuk memperbesar maupun memperkecil nilainya dengan mengatur ukuran roda gigi yang digunakan.

Salah satu jenis *belt* yang sering digunakan untuk permasalahan aplikasi perpindahan gerak adalah *timing belt*. Konstruksi *timing belt* dapat dilihat pada **Gambar 2.13**. *Timing belt* merupakan *belt* yang bertugas menyelaraskan dan menyalurkan putaran kruk as dengan noken as, sehingga proses buka dan tutup klep sesuai dengan posisi piston dan *timing* pengapian busi. *Timing belt* penting pada sebuah mesin, bila putus atau rusak mesin dapat seketika mati. Oleh karena fungsinya yang vital maka *timing belt* ini dibuat dari material yang lebih kuat. *Timing belt* memiliki gerigi – gerigi yang seragam, sehingga lebih baik dari pada menggunakan V-*belt* yang rawan terhadap selip. Pemilihan *timing belt*, terdapat beberapa spesifikasi yang perlu diperhatikan diantaranya gerigi yang lebih panjang akan mencegah adanya selip, konstruksi yang lebih ringan akan mengurangi rugi – rugi sentrifugal, daerah kontak (daerah pertemuan gerigi *belt* dan *gear*) yang lebih besar akan mengurangi tekanan unit, selain itu bagian persimpangan gerigi yang lebih besar akan menyebabkan kekuatan geser yang lebih besar.





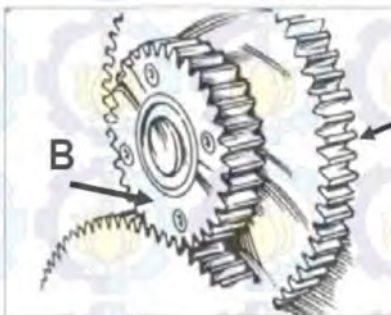
**Gambar 2.13** Konstruksi *Timing Belt* [9]

#### 2.2.4.1 Perbandingan Gear

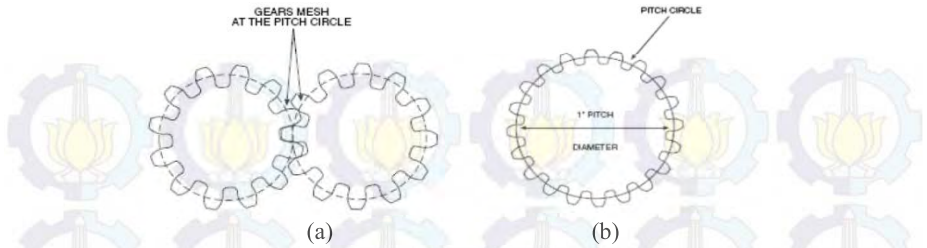
*Gear* merupakan suatu lingkaran yang memiliki gerigi di bagian luarnya. *Gear* ini digunakan untuk mengubah kecepatan sudut atau torsi. Terdapat dua konfigurasi dasar penggunaan *gear* yaitu pada satu poros dan beda poros.

##### a. Konfigurasi *Gear* Satu Poros (*Compound Gear*)

Jumlah pada *gear* dan *diameter pitch* merupakan dua hal yang harus diperhatikan dalam pemilihan kombinasi *gear* yang akan digunakan. Konfigurasi *gear* dengan sistem satu poros terlihat pada **Gambar 2.14**. Ketika dua buah *gear* bersinggungan terdapat lingkaran khayal yang disebut dengan *pitch circle* seperti pada **Gambar 2.15**. Diameter pada *pitch circle* adalah *pitch diameter*, dan rasio jumlah gigi terhadap *pitch diameter* adalah *pitch circle*.



**Gambar 2.14** Konfigurasi *Gear* Satu Poros [2]



**Gambar 2.15** (a) *Pitch Circle*, (b) *Gear* dengan 20 Gigi, *Pitch Diameter* 1'', dan *Diametral Pitch* 20 [2]

b. Konfigurasi *Gear* Beda Poros

*Gear* beda poros disebut juga dengan *gear* reduksi. Pada konfigurasi *gear* beda poros, *gear* menghubungkan dua poros yang berbeda dengan menggunakan *belt*, *chain*, ataupun rantai. Konfigurasi *gear* beda poros dapat dilihat pada **Gambar 2.16**. Konfigurasi dasar untuk dua *gear* yang berbeda poros dapat dilihat pada **Persamaan (2.4)**.

$$V_A \quad V_B \quad (2.4)$$

$$\omega_A R_A \quad \omega_B R_B$$

$$\frac{\omega_A}{\omega_B} \quad \frac{R_B}{R_A} \quad (2.5)$$

Misalkan  $R_A$   $R_B$ , maka

$$\omega_A \quad -\omega_B \quad (2.6)$$

Torsi pada *gear* dapat dihitung melalui **Persamaan (2.7)**.

$$\tau \quad F R \quad (2.7)$$

dengan  $\tau$  merupakan torsi (Nm),  $F$  merupakan gaya yang bekerja pada *gear* dan  $R$  merupakan jari-jari *gear*. Dari **Persamaan (2.6)** maka didapatkan aturan gaya yang bekerja pada *gear* adalah sebagai berikut :

$$F_A \quad F_B \quad (2.8)$$

$$\frac{\tau_A}{R_A} = \frac{\tau_B}{R_B}$$

$$\frac{\tau_A}{R_B} = \frac{\tau_B}{R_B}$$

sehingga didapatkan perbandingan torsi,

$$\frac{\tau_A}{\tau_B} = \frac{R_B}{R_B}$$

$$\tau_A \quad \tau_B \quad (2.9)$$

dengan,

$V_A$  : Kecepatan linier pada *gear* A

$V_B$  : Kecepatan linier pada *gear* B

$\omega_A$  : Kecepatan sudut pada *gear* A

$\omega_B$  : Kecepatan sudut pada *gear* B

$R_A$  : Jari-jari pada *gear* A

$R_B$  : Jari-jari pada *gear* B

$F_A$  : Gaya yang bekerja pada *gear* A

$F_B$  : Gaya yang bekerja pada *gear* B

$\tau_A$  : Torsi pada *gear* A

$\tau_B$  : Torsi pada *gear* B



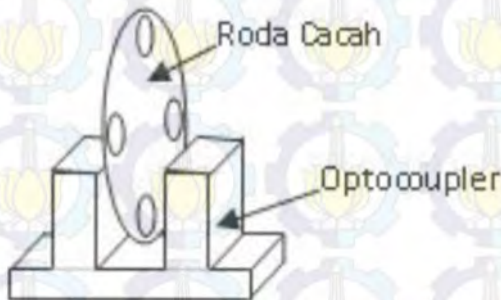
**Gambar 2.16** Konfigurasi *Gear* Beda Poros [2]



### 2.2.5 Rotary Encoder [10]

*Rotary encoder* dikenal sebagai salah satu sensor kecepatan. *Rotary encoder* adalah *device* elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* dapat dibuat dengan menggunakan sensor optik berupa *optocoupler* tipe “U” dan sebuah roda cacah berupa suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. *Optocoupler* merupakan komponen *optoisolator* yang memiliki karakteristik dimana penerima (*photo-transistor*) akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED inframerah) untuk penerima.

Prinsip kerja dari *rotary encoder* adalah ketika ada benda yang berada di antara celah sensornya, maka cahaya yang dikirimkan tidak bisa diterima oleh bagian penerima, sehingga menghasilkan tegangan keluaran yang nilainya mendekati VCC, begitu juga sebaliknya, jika tidak ada benda diantara celah sensornya maka akan menghasilkan tegangan keluaran yang nilainya mendekati 0 Volt. Sistem kerja dari *rotary encoder* adalah dengan mendeteksi adanya pulsa yang dihasilkan oleh terdeteksinya lubang pada lempengan yang dipasang pada poros. Pulsa yang dideteksi oleh sensor adalah berupa pulsa *high*. Banyaknya pulsa yang terdeteksi oleh sensor akan langsung dikonversi menjadi rpm dengan bantuan mikrokontroler. Konstruksi *rotary encoder* dapat dilihat pada **Gambar 2.17**.

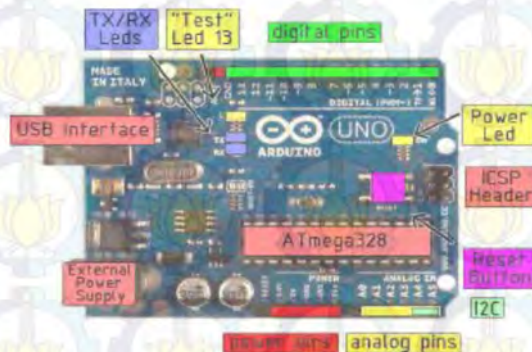


**Gambar 2.17** Konstruksi *Rotary Encoder* [10]

## 2.3 Sistem Minimum Arduino Uno R3 [11]

Arduino Uno adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino adalah sebuah *board* mikrokontroler yang berbasis ATmega328.

Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 *analog input*, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Arduino mampu *support* mikrokontroler; dan dapat dikoneksikan dengan komputer menggunakan kabel USB. *Board* Arduino Uno dapat dilihat pada **Gambar 2.18**.



**Gambar 2.18** *Board* Arduino Uno [11]

Arduino Uno berbeda dari semua *board* Arduino sebelumnya, Arduino Uno tidak menggunakan *chip driver* FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke *serial*. Revisi ke-2 dari *board* Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke *ground*, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode.

Arduino Uno memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, arduino yang lain, ataupun dengan mikrokontroler yang lain. ATmega328 yang digunakan pada Arduino Uno menyediakan komunikasi *serial* UART TTL (5V) yang tersedia pada pin *digital* 0 (RX) dan 1 (TX). Koneksi USB untuk menghubungkan dengan perangkat lain menggunakan ATmega16U2 yang mengacu sebagai *virtual com port* pada komputer. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. *Software* Arduino juga menyediakan *serial monitor* yang memungkinkan data teks dapat terkirim dan diterima oleh arduino dengan mudah. LED RX dan TX pada *board* Arduino akan

menyala saat data sedang dikirim atau diterima melalui kabel USB ke komputer (tetapi tidak menyala untuk komunikasi pada pin *digital 0* dan 1). Spesifikasi *board* Arduino Uno dapat dilihat pada **Tabel 2.2**.

**Tabel 2.2** Spesifikasi *Board* Arduino Uno [10]

Spesifikasi	Keterangan
Mikrokontroler	ATmega328
Tegangan Operasional	5 V
Tegangan <i>Input</i> (rekomendasi)	7-12 V
Tegangan <i>Input</i> (Limit)	6-20 V
Pin I/O digital	14 (6 diantaranya bisa digunakan untuk <i>output</i> PWM)
Pin Analog <i>Input</i>	6
Arus DC per Pin I/O	40 mA
Arus DC untuk pin 3.3 V	50 mA
<i>Flash Memory</i>	32 KB dengan 0,5 KB digunakan sebagai <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Kecepatan <i>Clock</i>	16 MHz

Arduino Uno dapat diprogram dengan *software* Arduino. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Lingkungan *open-source Arduino* memudahkan untuk menulis kode dan meng-*upload* ke *board* Arduino. *Software* Arduino dapat berjalan pada Windows, Mac OS X, dan Linux, dan perangkat lunak sumber terbuka lainnya.

## 2.4 Identifikasi Sistem [8]

Identifikasi sistem merupakan suatu proses untuk mendapatkan parameter-parameter dari suatu sistem berdasarkan hasil pengukuran *input output plant*. Berdasarkan hasil pengukuran *input output plant* tersebut, maka model matematis suatu *plant* dapat diketahui. Identifikasi dapat dilakukan pada sistem *closed loop* maupun *open loop*. Identifikasi *closed loop* lebih sulit dilakukan, namun dapat mengatasi sistem yang tidak pasti, tidak terpengaruh non linearitas, dan ketelitian tetap terjaga. Sedangkan identifikasi *open loop* lebih sederhana dan mudah dilakukan,

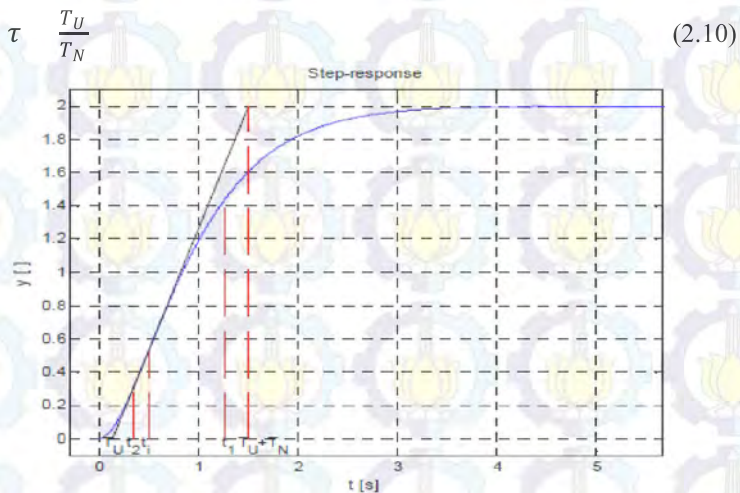


namun sulit diterapkan untuk sistem yang memiliki ketidakpastian respon. Identifikasi *open loop* merupakan salah satu identifikasi yang paling banyak digunakan. Dengan masukan *step*, *impuls*, atau *ramp*, maka respon transien dapat dicari melalui grafik respon sistem.

Pada Tugas Akhir ini identifikasi sistem dilakukan dengan menggunakan identifikasi Strejc untuk *plant* ICE dan identifikasi fisik dengan pengukuran parameter untuk *plant* motor DC.

### 2.4.1 Identifikasi Strejc [12]

Strejc merupakan salah satu metode identifikasi dengan penarikan garis singgung pada respon *plant*. Pada metode Strejc penarikan garis berdasar pada titik belok (*inflection point*) dari grafik respon seperti pada **Gambar 2.19**. Dengan ini didapatkan nilai  $T_U$  dan  $T_N$ .  $T_U$  merupakan waktu tunda sistem dan  $T_N$  merupakan waktu saat respon melewati waktu tunda hingga mencapai batas penarikan garis terhadap titik belok respon. Nilai  $T_U$  dan  $T_N$  berpengaruh terhadap orde *plant* dan untuk memodelkan respon *plant* berdasarkan tabel Strejc. Dengan diketahui nilai  $T_U$  dan  $T_N$ , maka dapat dicari nilai  $\tau$  berdasarkan **Persamaan (2.10)**. Metode Strejc melakukan pendekatan pada sistem yang tidak stabil tanpa *delay* (waktu tunda).



**Gambar 2.19** Respon *Step* Sistem dengan waktu  $t_1$ ,  $t_2$ ,  $t_b$ ,  $T_U$ , dan  $T_N$  [12]

1. Untuk  $\tau \geq$

Jika dari perbandingan didapatkan nilai  $\tau \geq$ , sistem didekati dengan orde ke-n dengan **Persamaan (2.11)** sebagai berikut:

$$G_{ST}(s) = \frac{K}{\tau_{ST} s^n} \quad (2.11)$$

Berdasarkan nilai  $\tau$  dapat ditentukan nilai  $y_i$  (dapat dilihat pada **Tabel 2.3**). Setelah didapatkan nilai  $y_i$ , maka melalui respon sistem dapat ditemukan nilai  $t_i$ . Jika nilai  $t_i$  sudah diketahui maka dapat ditemukan nilai  $\tau_{ST}$  seperti pada **Persamaan (2.12)**.

$$\tau_{ST} = \frac{t_i}{n-1} \quad (2.12)$$

**Tabel 2.3** Taksiran Orde ke-n dan Nilai Titik Singgung [12]

$n$	$\tau$	$y_i$	$n$	$\tau$	$y_i$
1	0,014	0,264	6	0,493	0,384
2	0,218	0,327	7	0,570	0,394
3	0,319	0,359	8	0,642	0,401
4	0,410	0,371	9	0,769	0,407
5	0,493	0,384	108	0,773	0,413

2. Untuk  $\tau <$

Jika dari perbandingan didapatkan nilai  $\tau <$ , sistem didekati dengan orde ke-n dengan **Persamaan (2.13)** sebagai berikut:

$$G_{ST} = \frac{K}{(\tau_{ST} s + 1) \tau_{ST} s} \quad (2.13)$$

Nilai  $\tau_{ST}$  dan  $\tau_{ST}$  dapat ditentukan dengan cara berikut:

- Mencari waktu  $t$  pada saat respon sistem mencapai 72% dari  $y_{ss}$ , kemudian tentukan jumlah dari  $\tau_{ST}$  dan  $\tau_{ST}$  berdasarkan **Persamaan (2.14)**.

$$\tau_{ST} + \tau_{ST} = \frac{t}{0.72} \quad (2.14)$$



- b. Menentukan parameter  $t$  dari **Persamaan (2.15)**.

$$t = \tau_{ST} - \tau_{ST} \quad (2.15)$$

- c. Menemukan nilai  $y(t)$  dari respon sistem berdasarkan **Gambar 2.19** dan menentukan perbandingan  $T$  untuk **Persamaan (2.16)** dari **Tabel 2.4**.

$$T = \frac{\tau}{\tau} \quad (2.16)$$

**Tabel 2.4** Nilai Konstan Perbandingan  $T$  [12]

$y$	$t_2$	$T$	$y$	$t_2$	$T$	$y$	$t_2$	$T$
0,30		0,000	0,25		0,105	0,20		0,264
0,29		0,023	0,24		0,128	0,19		0,322
0,28		0,043	0,23		0,154	0,18		0,403
0,27		0,063	0,22		0,183	0,17		0,538
0,26		0,084	0,21		0,219	0,16		1,000

#### 2.4.2 Identifikasi Fisik [13]

Untuk identifikasi secara fisik, biasanya identifikasi model dibagi berdasarkan identifikasi mekanik atau elektrik.

- a. Identifikasi Fisik Sistem Mekanik

Identifikasi fisik pada sistem mekanik menganalisa model matematis berdasar komponen pendukungnya. Hukum dasar yang digunakan dalam identifikasi sistem mekanik adalah hukum Newton kedua yang hanya dapat diaplikasikan pada sistem mekanik. Dengan rumus  $F = m \cdot a$  dapat diartikan untuk perhitungan seluruh resultan gaya yang bekerja pada suatu sistem sama dengan turunan dari momentum linier benda tersebut terhadap waktu

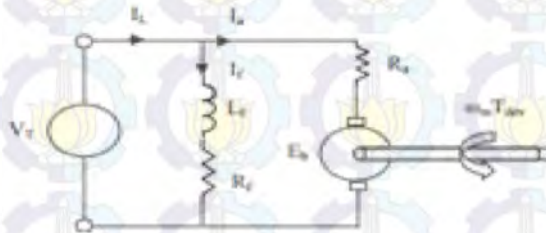
- b. Identifikasi Fisik Sistem Elektrik

Pada sistem elektrik, hukum dasar yang digunakan adalah hukum Kirchhoff arus dan tegangan. Hukum Kirchhoff arus (hukum *node*/titik) berbunyi bahwa penjumlahan seluruh arus yang masuk dan keluar adalah nol atau dapat dikatakan bahwa jumlah seluruh arus yang masuk dan seluruh arus yang keluar adalah sama. Hukum Kirchhoff tegangan (hukum *loop*) berbunyi bahwa penjumlahan

seluruh tegangan dalam satu *loop* pada sirkuit elektrik adalah nol atau dapat dikatakan bahwa jumlah seluruh tegangan naik sama dengan jumlah seluruh tegangan turun pada satu *loop* tertutup. Model matematis dari sebuah sirkuit elektrik dapat ditentukan dengan menggunakan salah satu atau kedua hukum Kirchhoff tersebut.

#### 2.4.2.1 Identifikasi Fisik Motor DC Konfigurasi Paralel

Konfigurasi motor DC paralel secara fisik dapat dilihat pada **Gambar 2.20**.



**Gambar 2.20** Konfigurasi Fisik Motor DC Konfigurasi Paralel

Motor DC dengan konfigurasi paralel memiliki karakteristik non linear terhadap arus jangkar pada **Persamaan (2.17)**, arus medan pada **Persamaan (2.18)**, dan torsi motor pada **Persamaan (2.19)**. Dari konfigurasi pada **Gambar 2.20** dapat dicari pemodelan fisik motor DC paralel sebagai berikut :

a. Rangkaian Arus Medan

$$\begin{aligned} I_f &= \frac{R_f}{L_f s + R_f} E_a \\ L_f \dot{I}_f &= R_f I_f + E_a \\ \dot{I}_f &= -\frac{R_f}{L_f} I_f + \frac{1}{L_f} E_a \end{aligned} \quad (2.17)$$

b. Rangkaian Arus Jangkar

$$I_a = \frac{1}{L_a s + R_a} E_a - E_{ggl}$$

$$L_a \dot{I}_a = R_a I_a - E_a - E_{ggl}$$

$$\dot{I}_a = -\frac{R_a}{L_a} I_a - \frac{1}{L_a} E_a - \frac{1}{L_a} E_{ggl}$$

$$E_{ggl} = K_{ggl} I_f \Omega_m, \text{ sehingga :}$$

$$\dot{I}_a = -\frac{R_a}{L_a} I_a - \frac{1}{L_a} E_a - \frac{1}{L_a} K_{ggl} I_f \Omega_m \quad (2.18)$$

### c. Rangkaian Torsi Motor

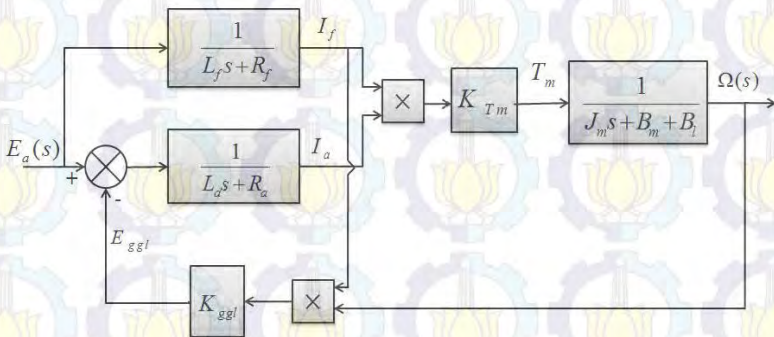
$$\Omega_m = \frac{T_m - T_l}{J_m s + B_m}$$

$$(J_m s + B_m) \Omega_m = T_m - B_l \Omega_m$$

$$J_m \dot{\Omega}_m + B_m \Omega_m = K_{Tm} I_a I_f - B_l \Omega_m$$

$$\dot{\Omega}_m = \frac{K_{Tm} I_a I_f}{J_m} - \frac{B_m \Omega_m}{J_m} - \frac{B_l \Omega_m}{J_m} \quad (2.19)$$

**Persamaan (2.17)** merupakan persamaan non linear motor DC terhadap arus medan, **Persamaan (2.18)** merupakan persamaan non linear motor DC terhadap arus jangkar, dan **Persamaan (2.19)** merupakan persamaan non linear motor DC terhadap torsi motor. Persamaan-persamaan diatas tersebut dapat direpresentasikan dalam diagram blok motor DC paralel seperti pada **Gambar 2.21**.



**Gambar 2. 21** Blok Diagram Fisik Motor DC Konfigurasi Paralel



## 2.5 Representasi *State Space* [13]

Dalam sistem pengaturan modern, sistem yang akan diatur menjadi lebih kompleks, terutama karena adanya persyaratan untuk mencapai akurasi yang tinggi meskipun sistem sangat kompleks. Sistem yang kompleks dapat memiliki lebih dari satu *input* dan *output*. Karena perlunya memenuhi persyaratan yang semakin ketat pada kinerja pengaturan sistem, peningkatan kompleksitas sistem dan kemudahan akses dalam skala besar, maka dikembangkan teori pengaturan modern yang merupakan pendekatan baru untuk analisis dan desain sistem kontrol yang kompleks. Teori modern ini dikembangkan sejak sekitar tahun 1960 dan didasarkan pada konsep *state*. Pendekatan baru yang kemudian disebut analisis *state space* ini kemudian menjadi teori baru yang marak digunakan pada sistem yang dinamik dan memiliki banyak kompleksitas.

Pada analisis *state space*, terdapat tiga variabel yang harus diperhatikan pada permodelan sistem dinamik yaitu variabel *input*, variabel *output* dan variabel *state*. Persamaan *state space* dapat diasumsikan pada sistem *Multiple Input Multiple Output* (MIMO) dengan  $n$  *integrator*. Dimana dapat diasumsikan terdapat  $r$  input yaitu  $u(t)$   $u(t) \dots \dots u_r(t)$  dan  $m$  output yaitu  $y(t)$   $y(t) \dots \dots y_m(t)$ . Tetapkan  $n$  *output* dari *integrator* sebagai variabel *state*  $x(t)$   $x(t) \dots \dots x_n(t)$ , maka sistem dapat dituliskan pada **Persamaan (2.20)**.

$$\begin{aligned} \dot{x}_1(t) &= f_1 x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \\ \dot{x}_2(t) &= f_2 x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \\ &\vdots \\ \dot{x}_n(t) &= f_n x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \end{aligned} \quad (2.20)$$

*output* sistem dapat dituliskan pada **Persamaan (2.21)**.

$$\begin{aligned} y_1(t) &= g_1 x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \\ y_2(t) &= g_2 x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \\ &\vdots \\ y_n(t) &= g_n x_1 x_2 \dots x_n u_1 u_2 \dots u_r t \end{aligned} \quad (2.21)$$

jika ditentukan,

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix},$$

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, \mathbf{g}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix},$$

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}$$

Lalu **Persamaan (2.20)** dan **Persamaan (2.21)** dapat dituliskan sebagai :

$$\dot{\mathbf{x}}(t) = \mathbf{f} \mathbf{x} \mathbf{u} t \quad (2.22)$$

$$\mathbf{y}(t) = \mathbf{g} \mathbf{x} \mathbf{u} t \quad (2.23)$$

**Persamaan (2.22)** merupakan persamaan *state* dan **Persamaan (2.23)** merupakan persamaan *output*. Jika kedua persamaan tersebut dilinearisasikan dengan menggunakan operasi *state* dan vektor **f** dan **g** berdasar waktu, maka akan didapat persamaan linier dari persamaan *state* pada **Persamaan (2.24)** dan persamaan *output* pada **Persamaan (2.25)**.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u} t \quad (2.24)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u} t \quad (2.25)$$

Pada sistem dinamik dengan fungsi penghantar, representasi *state space* dapat dijelaskan dengan **Persamaan (2.26)**

$$\frac{Y}{U} = \frac{s^n \quad b \quad s^{n-1} \quad b \quad s^{n-2} \quad \dots \quad b_{n-1} \quad s \quad b_n}{s^n \quad a \quad s^{n-1} \quad \dots \quad a_{n-1} \quad s \quad a_n} \quad (2.26)$$

atau dapat dituliskan dalam bentuk persamaan diferensial pada **Persamaan (2.27)**,

$$y^n + a y^{n-1} + \dots + a_n y = b u^n + b u^{n-1} + \dots + b_n u \quad (2.27)$$

lalu dengan memilih **Persamaan(2.28)**,

$$\begin{aligned} x &= y - \beta u \\ \dot{x} &= \dot{y} - \beta \dot{u} - \beta u \quad \dot{x} = -\beta u \\ \ddot{x} &= \ddot{y} - \beta \ddot{u} - \beta \dot{u} - \beta u \quad \ddot{x} = -\beta u \\ &\vdots \\ x_{n-1} &= y^{(n-1)} - \beta u^{(n-1)} - \dots - \beta_{n-1} u \quad \dot{x}_{n-1} = -\beta_{n-1} u \\ x_n &= y^{(n)} - \beta u^{(n)} - \dots - \beta_{n-1} u \quad \dot{x}_n = -\beta_{n-1} u \end{aligned} \quad (2.28)$$

dengan  $\beta, \beta, \beta, \dots, \beta_n$  ditentukan melalui **Persamaan (2.29)** :

$$\begin{aligned} \beta &= b \\ \beta &= b - a \beta \\ \beta &= b - a \beta - a \beta \\ &\vdots \\ \beta_{n-1} &= b_{n-1} - a \beta_{n-1} - a_{n-1} \beta \\ \beta_n &= b_n - a \beta_{n-1} - \dots - a_{n-1} \beta - a_n \beta \end{aligned} \quad (2.29)$$

maka didapatkan **Persamaan (2.30)** :

$$\begin{aligned} \dot{x} &= x - \beta u \\ \dot{x} &= x - \beta u \\ &\vdots \\ \dot{x}_{n-1} &= x_{n-1} - \beta_{n-1} u \\ \dot{x} &= -a_n x - a_{n-1} x - \dots - a x_n - \beta_{n-1} u \end{aligned} \quad (2.30)$$

dan jika dilihat dalam bentuk persamaan vektor matriks, **Persamaan (2.30)** dapat dituliskan sebagai berikut :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \beta_0 u$$

atau :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (2.31)$$

$$y = \mathbf{C}\mathbf{x} + \mathbf{D}u \quad (2.32)$$

dengan:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \mathbf{D} = \beta_0 = b_0$$

Persamaan *state space* tidak harus mengambil variabel dari *output* sistem, tetapi dapat juga menggunakan variabel lain seperti *error* atau kombinasi linier lainnya.



## 2.6 Root Mean Square Error [2]

Untuk mengevaluasi terhadap perkiraan *error* digunakan komponen dari sebuah model statistika yang disebut RMSE. Carbone dan Armstrong (1982) menemukan bahwa *Root Mean Square Error* merupakan perhitungan perkiraan akurasi dari pemodelan sistem yang paling diminati. RMSE mengukur akurasi pada nilai deret waktu secara statistik seperti halnya regresi. RMSE dapat merepresentasikan ukuran dari *error* rata-rata karena RMSE membandingkan hasil data pengukuran dan data pemodelan pada skala yang sama antara kedua data tersebut. Formulasi perhitungan RMSE dapat dilihat pada **Persamaan (2.33)** dan formulasi nilai *error* dalam bentuk presentasi dapat dilihat pada **Persamaan (2.34)**.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (2.33)$$

$$e_i = \frac{A_i - M_i}{A_i} \quad (2.34)$$

dengan,

- n : Jumlah data
- i : Urutan data
- e : Nilai error
- A : Nilai data hasil pengukuran
- M : Nilai data hasil pemodelan

Pemodelan dengan menggunakan RMSE tidak dapat secara mutlak dikatakan terbaik atau sesuai dengan model dinamika sistem, namun untuk menentukan pemodelan yang mirip dengan data pengukuran dapat dilihat dengan nilai RMSE yang terkecil dari perhitungan lainnya.

## 2.7 Kontroler [14]

Dalam sebuah sistem kontrol, kontroler mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem, yaitu kontroler. Kontroler merupakan salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*set point*) atau sinyal *error*



mejadi sinyal kontrol. Sinyal *error* disini adalah selisih antara sinyal umpan balik yang dapat berupa sinyal keluaran *plant* sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (*set point*). Salah satu tugas komponen kontroler adalah meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal *set point* dan sinyal aktual. Hal ini sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang senantiasa (diinginkan) sama dengan sinyal *set point*. Semakin cepat reaksi sistem (sinyal aktual) mengikuti sinyal *set point* dan semakin kecil kesalahan yang terjadi, semakin baiklah kinerja sistem kontrol yang diterapkan.

### 2.7.1 Kontroler *Proportional Integral Derivative* (PID) [14]

Salah satu jenis kontroler berdasarkan aksi kontrolnya adalah kontroler PID. Kontroler PID merupakan kontroler yang aksi kontrolnya merupakan kombinasi dari aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol differensial. Untuk kontroler PID, sinyal kesalahan  $e(t)$  merupakan masukan kontroler sedangkan keluaran kontroler adalah sinyal kontrol  $u(t)$ . Hubungan antara masukan kontroler  $e(t)$  dan keluaran kontroler  $u(t)$  dapat dilihat pada **Persamaan (2.35)**.

$$u(t) = K_p \left\{ e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de}{dt} \right\} \quad (2.35)$$

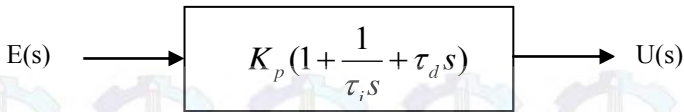
atau dalam besaran transformasi Laplace :

$$U(s) = K_p \left( \frac{1}{\tau_i s} + \tau_d s \right) E(s) \quad (2.36)$$

$K_p$  adalah penguatan proporsional,  $\tau_i$  adalah waktu integral dan  $\tau_d$  adalah waktu differensial. Parameter  $K_p$ ,  $\tau_i$ , dan  $\tau_d$  ketiganya dapat ditentukan. Sehingga fungsi alih kontroler PID seperti pada **Persamaan (2.37)**.

$$\frac{U(s)}{E(s)} = K_p \left\{ \frac{1}{\tau_i s} + \tau_d s \right\} \quad (2.37)$$

Diagram blok kontroler PID adalah seperti pada **Gambar 2.22** sebagai berikut :



**Gambar 2.22** Diagram Blok Kontroler PID

Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi dari ketiga parameter  $P$ ,  $I$  dan  $D$ . Pengaturan konstanta  $K_p$ ,  $K_i$ ,  $K_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diatur lebih menonjol dibanding yang lain. Pengaruh parameter  $K_p$ ,  $K_i$ , dan  $K_d$  terhadap respon dapat dilihat pada **Tabel 2.5**.

**Tabel 2.5** Pengaruh Parameter  $K_p$ ,  $K_i$ , dan  $K_d$

<i>Closed-Loop Response</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Error Steady State</i>
$K_p$	<i>Decrease</i>	<i>Increase</i>	<i>Small Change</i>	<i>Decrease</i>
$K_i$ $K_p \frac{1}{\tau_i}$	<i>Decrease</i>	<i>Increase</i>	<i>Increase</i>	<i>Eliminate</i>
$K_d$ $K_p \tau_d$	<i>Small Change</i>	<i>Decrease</i>	<i>Decrease</i>	<i>Small Change</i>

### 2.7.2 Kontroler Optimal *Linear Quadratic Regulator* (LQR) [15] [16]

*Linear Quadratic Regulator* adalah suatu kontrol optimal pada sistem linear dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator (*Regulator Problem*). Dalam pengaturan optimal ada 3 hal yang dapat dijadikan acuan untuk merancang kontroler, yaitu :

1. Meminimumkan waktu
2. Meminimumkan kesalahan (*error*)
3. Meminimumkan energi kontrol

Pada Tugas Akhir ini dipilih metode pengaturan optimal yang dapat meminimumkan kesalahan (*error*) ketika terjadi pembebanan.

Desain kontrol optimal dilakukan melalui formulasi variabel *state*. Pada dasarnya solusi kontrol optimal dititikberatkan pada pencarian nilai sinyal kontrol optimal  $u(t)$ , sehingga indeks performansi dioptimasi.

Suatu *plant* linear dalam *state space* pada **Persamaan (2.38)** dinyatakan sebagai berikut:

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (2.38)$$

dengan,

$\mathbf{A}$  : nxn matriks *state*

$\mathbf{B}$  : nxr matriks *control*

$\mathbf{x}(t)$  : nth vektor *state*

$\mathbf{u}(t)$  : rth vektor *control*

Adapun indeks performansi dalam interval  $[0, \infty]$  secara matematis dinyatakan pada **Persamaan (2.39)** sebagai berikut :

$$J = \int_0^{\infty} [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)]dt \quad (2.39)$$

dengan,

$t_0$  : waktu awal

$\infty$  : waktu akhir (*infinite*)

$\mathbf{Q}$  : matriks nxn simetri, semidefinit positif

$\mathbf{R}$  : matriks mxm simetri, definit positif

Metode kontrol optimal bertujuan untuk menjaga agar *state*  $\mathbf{x}(t)$  mendekati 0 yang disebut *state regulator system*. Dengan kata lain, untuk mendapatkan sinyal kontrol  $\mathbf{u}(t)$ , apabila harga keluaran awal tidak nol, maka masukannya harus ditentukan agar keluarannya menjadi nol.

Kontrol optimal diberikan sebagai berikut :

$$\mathbf{u}^*(t) = -\mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}}\mathbf{x}^*(t) - \mathbf{K}\mathbf{x}^* \quad (2.40)$$

dengan Kalman *gain*,

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}} \quad (2.41)$$



$\bar{P}$ ,  $n \times n$  simetri, definit positif, matriks konstan, dan merupakan solusi dari nonlinier dengan persamaan *Algebraic Riccati Equation* (ARE) yang dituliskan sebagai berikut :

$$A \bar{P} + \bar{P} A - \bar{P} B R^{-1} B^T \bar{P} = -Q \quad (2.42)$$

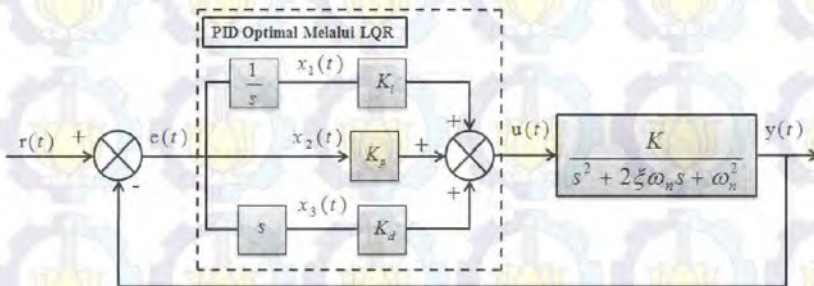
Penentuan kontrol objektif pada sistem kontrol optimal dan menentukan matriks pembobotan  $Q$  dan  $R$ , belum ada solusi yang unik dari matriks-matriks tersebut. Untuk menentukan matriks pembobotan tersebut biasanya dilakukan dengan *trial and error* (coba-coba) di mana pemilihan pemberat  $Q$  dan  $R$  berpedoman pada:

- Semakin besar harga  $Q$ , semakin memperbesar harga elemen penguatan  $K$  sehingga mempercepat sistem untuk mencapai *steady state*.
- Semakin besar harga  $R$ , maka akan memperkecil harga penguatan  $K$  dan memperlambat sistem mencapai *steady state*.

### 2.7.3 Kontroler PID-LQR [17] [18]

Masalah utama kontroler PID adalah *tuning*, yaitu menentukan nilai parameter  $K_p$ ,  $K_i$ , dan  $K_d$  agar diperoleh performansi sistem yang optimal. Umumnya proses *tuning* parameter PID masih dilakukan secara manual, sehingga memakan waktu. Yu mengembangkan metode LQR (*Linear Quadratic Regulator*) untuk *tuning* PID dengan hasil yang optimal, tetapi perlu kalkulasi matematika dan penyelesaian yang rumit.

Metode PID-LQR merupakan metode PID optimal dengan pendekatan LQR untuk menentukan parameter  $K_p$ ,  $K_i$  dan  $K_d$  pada kontroler PID. Diagram blok PID-LQR dapat ditunjukkan pada **Gambar 2.23**.



**Gambar 2.23** Diagram Blok PID-LQR

Pada metode PID-LQR terdapat beberapa prosedur yang dilakukan untuk mendapatkan parameter kontroler PID. Langkah-langkah perancangan sistem kontrol PID-LQR dapat dijelaskan sebagai berikut :

1. Fungsi alih yang didapatkan dari proses identifikasi *plant* harus diubah dalam representasi *state space*. Diketahui bahwa *plant* merupakan orde dua dengan *input*  $u(t)$ , *output*  $y(t)$ , dan fungsi alih *plant* seperti pada **Persamaan (2.43)**.

$$\frac{Y(s)}{U(s)} = \frac{a}{s^2 + \xi \omega_n s + \omega_n^2} = \frac{-E s}{U s} \quad (2.43)$$

Selanjutnya dimisalkan sebuah *state* :

$$\dot{x} = \int e(t) dt \quad \dot{x} = e(t) \quad \dot{x} = \frac{de}{dt} \quad (2.44)$$

Representasi *plant* dalam *state space* dapat dibentuk sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -\xi \omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -a \end{bmatrix} u \quad (2.45)$$

sehingga diperoleh parameter *plant* sebagai berikut :

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -\xi \omega_n \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ -a \end{bmatrix} \quad (2.46)$$

2. Menentukan matriks pembobot **Q** dan **R**  
Matriks **Q** merupakan koefisien pembobot yang digunakan untuk menentukan lebar area *state*, sedangkan matriks **R** digunakan untuk menentukan lebar area sinyal kontrol **u**.
3. Menyelesaikan persamaan *Algebraic Riccati Equation* (ARE).  
Persamaan *Algebraic Riccati Equation* (ARE) seperti ditunjukkan pada **Persamaan (2.47)** :

$$\mathbf{A} \bar{\mathbf{P}} + \bar{\mathbf{P}} \mathbf{A} - \bar{\mathbf{P}} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} + \mathbf{Q} = 0 \quad (2.47)$$

Penyelesaian ini digunakan untuk memperoleh matriks Riccati **P** dengan bantuan matriks **A**, matriks **B**, dan matriks pembobot **Q** dan **R**.

4. Mendapatkan parameter PID berdasarkan persamaan Kalman *gain*. Persamaan Kalman *gain* ditunjukkan pada **Persamaan (2.48)** sebagai berikut :

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} \quad (2.48)$$

$$\mathbf{K} = \mathbf{R}^{-1} \begin{bmatrix} -a \begin{bmatrix} P_1 & P_1 & P_1 \\ P_1 & P_1 & P_1 \\ P_1 & P_1 & P_1 \end{bmatrix} \end{bmatrix} \quad (2.49)$$

$$\mathbf{K} = -\mathbf{R}^{-1} \begin{bmatrix} P_1 & P_1 & P_1 \\ P_1 & P_1 & P_1 \\ P_1 & P_1 & P_1 \end{bmatrix} \quad (2.50)$$

$$\mathbf{K} = -[aR^{-1} P_1 \quad : aR^{-1} P_1 \quad : aR^{-1} P_1] \quad (2.51)$$

$$\mathbf{K} = -[K_i \quad : K_p \quad : K_d] \quad (2.52)$$

Kemudian sinyal kontrol optimal solusi PID-*Linear Quadratic Regulator* dapat direpresentasikan ke dalam **Persamaan (2.53)** sebagai berikut :

$$\mathbf{u}^*(t) = -\mathbf{K} \mathbf{x}^*(t) \quad (2.53)$$

$$\mathbf{u}^*(t) = K_i x(t) + K_p x(t) + K_d x(t) \quad (2.54)$$

$$\mathbf{u}^*(t) = K_i \int e(t) dt + K_p e(t) + K_d \frac{de(t)}{dt} \quad (2.55)$$



## BAB 3

### PERANCANGAN SISTEM

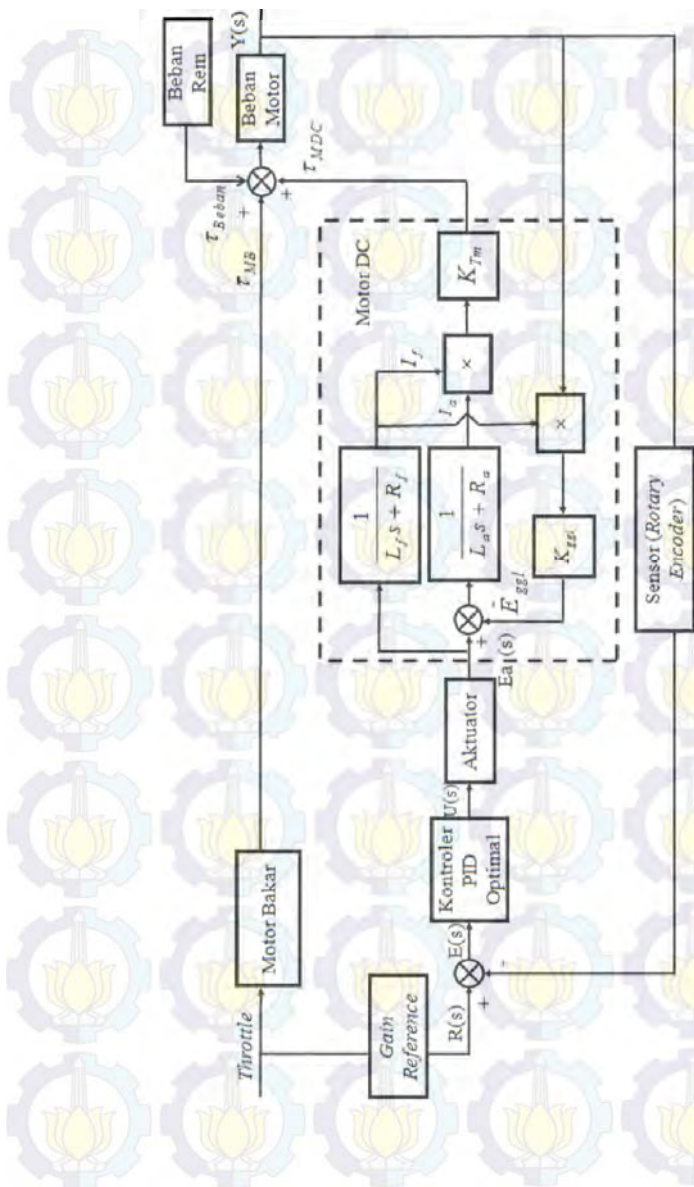
#### 3.1 Gambaran Umum Sistem

HEV merupakan kendaraan listrik hibrid yang bekerja dengan mengkombinasikan pemanfaatan mesin bakar dan motor listrik sebagai tenaga penggerak. HEV memiliki beberapa konfigurasi yaitu konfigurasi seri, konfigurasi paralel, dan konfigurasi seri-paralel. Pada Tugas Akhir ini digunakan Simulator *Hyrid Electric Vehicle* dengan konfigurasi paralel. Prinsip kerja HEV dengan konfigurasi paralel adalah mesin bakar dan motor listrik dapat menjalankan kendaraan secara bersama-sama dengan mesin bakar sebagai penggerak utama dan motor DC sebagai penggerak pembantu. Pada konfigurasi paralel peran motor DC sangat penting untuk membantu mesin bakar. Ketika mesin bakar sebagai penggerak utama diberikan efek beban yang berlebih, maka kecepatan putar yang dihasilkan mesin bakar akan berkurang. Sebagai penggerak pembantu, Motor DC akan memberikan bantuan sehingga sesuai dengan kecepatan putar yang diinginkan. Pengujian ini dilakukan dengan menggunakan rem elektromagnetik sebagai representasi efek beban untuk diberikan kepada mesin bakar ketika telah mencapai kecepatan stabil. Dengan demikian dapat diketahui fungsi motor DC untuk membantu kerja dari mesin bakar.

Simulator HEV yang digunakan pada Tugas Akhir ini merupakan hasil rancangan dari Wahyu Sriwidodo dan Galih Satriyo A.W. pada Tugas Akhir pada tahun 2010 seperti terlihat pada **Gambar 3.1**. Diagram blok sistem dapat dilihat pada **Gambar 3.2**.



**Gambar 3.1** Simulator HEV dengan Konfigurasi Paralel



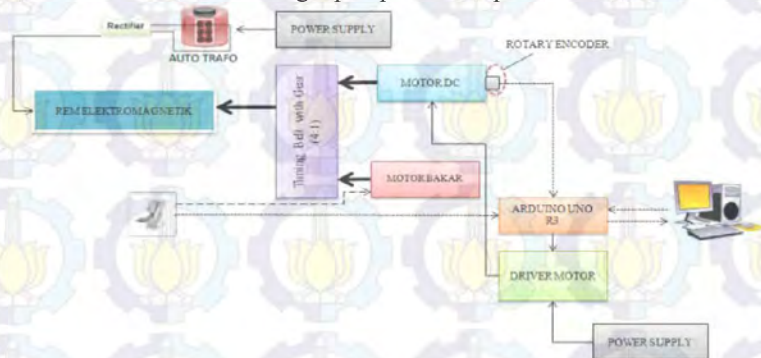
Gambar 3.2 Diagram Blok Sistem



Pada Tugas Akhir ini digunakan metode kontrol berbasis kontrol optimal yaitu *PID-Linear Quadratic Regulator*. Metode kontrol tersebut digunakan untuk mengatur sinyal kontrol yang diperlukan untuk mengatur kecepatan motor DC sehingga ketika diberi beban berlebih, kecepatan putar sesuai dengan yang diharapkan.

### 3.2 Perancangan Perangkat Keras

Secara garis besar simulator PHEV merupakan gabungan antara ICE dan motor DC, selain itu juga terdapat perangkat pendukung lainnya yang bekerja secara sinergi. Konfigurasi perangkat keras simulator PHEV secara lengkap dapat dilihat pada **Gambar 3.3**.



**Gambar 3.3** Konfigurasi Perangkat Keras Simulator PHEV

Pada konfigurasi HEV secara lengkap terdapat motor DC, mesin bakar, rem elektromagnetik, sistem minimum Arduino, dan *rotary encoder*. Selain itu, untuk mengubah tegangan AC menjadi tegangan DC diperlukan rangkaian penyearah gelombang penuh sebagai tegangan masukan motor DC. Dilihat dari spesifikasi, putaran dari mesin bakar jauh lebih besar dari motor DC sehingga kecepatan putar ICE harus dikurangi secara proporsional. Oleh karena itu diperlukan transmisi gir agar kecepatan putar pada ICE sebanding dengan motor DC.

Potensiometer juga diperlukan pada perancangan dan dipasang pada pedal gas untuk merepresentasikan bukaan *throttle* yang merupakan masukan pada ICE, terletak pada karburator dan terhubung dengan pedal gas. Sedangkan keluaran sistem diukur menggunakan *rotary encoder*. Keluaran pada sistem merupakan kecepatan putar pada poros motor DC sebagai poros utama yang akan menanggung beban

berupa rem elektromagnetik yang menghasilkan torsi lawan terhadap mesin bakar. Selain itu juga diperlukan sistem minimum Arduino dan rangkaian *driver* untuk sistem pengaturannya.

Perancangan perangkat keras untuk simulator HEV dalam Tugas Akhir ini dapat dibagi menjadi 2 bagian yaitu perangkat mekanik dan perangkat elektronik

### 3.2.1 Perancangan Perangkat Mekanik

Terdapat beberapa perangkat dalam perancangan mekanik untuk memenuhi suatu sistem seperti konfigurasi ICE dan motor DC pada bagian perangkat utama dan juga perangkat pendukung lainnya.

#### 3.2.1.1 *Internal Combustion Engine (ICE)*

Pada simulator HEV yang digunakan dalam Tugas Akhir ini digunakan mesin pemotong rumput. Mesin pemotong rumput direpresentasikan sebagai ICE yang berperan sebagai penggerak utama untuk menanggung beban. Mesin pemotong rumput yang digunakan adalah jenis mesin bakar 2 Tak. Mesin pemotong rumput 2 tak dipilih karena memiliki kecepatan putar yang tinggi dan poros engkol terhubung secara langsung dengan poros keluaran tanpa ada sistem transmisi. Spesifikasi mesin pemotong rumput 2 Tak dapat dilihat pada **Tabel 3.1**. Bentuk fisik mesin pemotong rumput 2 Tak yang digunakan pada simulator HEV dapat dilihat pada **Gambar 3.4**.

**Tabel 3.1** Spesifikasi Mesin Pemotong Rumput 2 Tak

No	Klasifikasi	Spesifikasi
1	Jenis	2 Tak
2	Bahan Bakar	Bensin dan Oli Samping
3	Perbandingan Bensin dan Oli	25 : 1
4	Kapasitas Silinder	33.8 cc
5	Pemasukan Bahan Bakar	Karburator
6	Starter	Recoil Starter
7	Sistem Pengapian	Electronic Ignition
8	Sistem Kopling	Kering, Sentrifugal
9	Kecepatan Putar Maks	8000 rpm
10	Daya Keluaran	1.8 HP



**Gambar 3.4** Bentuk Fisik Mesin Pemotong Rumput 2 Tak pada Simulator HEV

### 3.2.1.2 Motor Listrik

Pada simulator HEV digunakan motor listrik jenis motor DC. Motor DC merupakan motor yang mengubah energi listrik DC menjadi energi gerak. Pada simulator HEV, motor DC berfungsi sebagai penggerak pembantu, oleh karena itu dipilih spesifikasi motor DC yang memiliki daya keluaran lebih kecil daripada ICE. Selain itu, Motor DC dipilih karena motor DC lebih mudah dikendalikan dibandingkan dengan motor *Alternating Current* (AC). Pada Tugas Akhir ini, motor DC akan diatur agar dapat membantu kerja ICE ketika menanggung beban. Spesifikasi motor DC yang digunakan dapat dilihat pada **Tabel 3.2**. Bentuk fisik motor DC yang digunakan pada simulator HEV dapat dilihat pada **Gambar 3.5**.

**Tabel 3.2** Spesifikasi Motor DC

No	Klasifikasi	Spesifikasi
1	Tegangan Maksimal	190 V
2	Kecepatan Putar Maks	1750 rpm
3	Tipe	Seri/Paralel
4	Daya Keluaran	0.5 HP





**Gambar 3.5** Bentuk Fisik Motor DC pada Simulator HEV

Motor DC yang digunakan merupakan motor DC universal, lalu pada simulator HEV motor DC dikonfigurasi secara paralel.

### 3.2.1.3 Roda Gigi Penghubung

Pada simulator HEV ini, poros ICE dan poros motor DC sebagai poros utama yang terhubung dengan beban terpasang secara terpisah. Untuk menyatukan poros ICE dan poros motor DC dibutuhkan pemasangan *timing belt* dan *gear*. Dengan demikian, ICE dapat menyalurkan daya ke poros utama. Sebagaimana diketahui bahwa putaran ICE lebih besar daripada putaran motor DC, sehingga perlu adanya transmisi *gear* yang dapat mereduksi kecepatan ICE secara proporsional sehingga sesuai dengan rentang kecepatan putar motor DC.

Perbandingan roda gigi digunakan untuk menyamakan rentang kecepatan putar antara keduanya. Rentang kerja ICE antara 0 – 8000 rpm dan rentang kerja motor DC pada 0 – 2000 rpm, dengan menggunakan teori perbandingan roda gigi dapat dicari perbandingan *gear* yang harus digunakan dengan rumus seperti pada **Persamaan (3.1)**.

$$\omega_a R_a = \omega_b R_b \quad (3.1)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_a} \quad (3.2)$$

$$\frac{R_b}{R_a}$$

$$\frac{R_a}{R_b} = \quad (3.3)$$

dengan,

$\omega_a$  : Kecepatan putar di motor DC

$\omega_b$  : Kecepatan putar di ICE

$R_a$  : Jari-jari gir pada poros motor DC

$R_b$  : Jari-jari gir pada poros ICE

Berdasarkan persamaan didapatkan perbandingan roda gigi 1 : 4 untuk menyamakan kecepatan putar antara ICE dan motor DC. Akan tetapi, secara fisik perbandingan roda gigi 1:4 tidak efisien dari segi tempat. Untuk efisiensi tempat, maka perbandingan *gear* dibagi menjadi dua bagian dengan memberikan *gear* tambahan di antara keduanya seperti terlihat pada **Gambar 3.6**.



**Gambar 3.6** Transmisi *Gear*

Perbandingan  $R_{c1} : R_{c2} = 2:1$ , *gear* poros ICE  $R_b$  digandeng dengan *gear* poros tengah  $R_{c1}$  dengan perbandingan  $R_b : R_{c1} = 1:2$ . *Gear* poros tengah  $R_{c2}$  digandeng dengan *gear* poros motor DC  $R_a$  dengan perbandingan  $R_{c2} : R_a = 1:2$ , sehingga perhitungan perbandingan *gear* yang digunakan pada simulator HEV adalah seperti pada **Persamaan (3.4)** sebagai berikut :

$$\begin{aligned} \omega_a R_a &= \omega_c R_c \\ \omega_c &= \frac{\omega_a R_a}{R_c} \\ \omega_b R_b &= \omega_c R_c \end{aligned} \quad (3.4)$$

$$\omega_c = \frac{\omega_b R_b}{R_c} \quad (3.5)$$

dengan,

$\omega_c$  : Kecepatan putar pada poros tengah 1

$\omega_c$  : Kecepatan putar pada poros tengah 2

$R_c$  : Jari-jari gir pada poros tengah 1

$R_c$  : Jari-jari gir pada poros tengah 2

karena  $\omega_c = \omega_c$ , maka :

$$\frac{\omega_a R_a}{R_c} = \frac{\omega_b R_b}{R_c} \quad (3.6)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_c} \cdot \frac{R_c}{R_a}$$

$$\frac{\omega_a}{\omega_b} = - - - \quad (3.7)$$

Berdasarkan pada persamaan, dapat dilihat bahwa apabila  $\omega_a$  diberikan nilai 2000 rpm maka akan didapatkan nilai  $\omega_b$  sebesar 8000 rpm. Sehingga dengan konfigurasi penyatuan poros akan menghasilkan nilai reduksi yang sama dengan menggunakan perbandingan *gear* secara langsung.

Selain kecepatan putar, torsi juga akan mengalami perubahan dengan adanya reduksi ini. Hubungan antara perbandingan torsi dengan jari-jari *gear* dapat dilihat pada **Persamaan (3.8)**.

$$\frac{\tau_a}{R_a} = \frac{\tau_b}{R_b} \quad (3.8)$$

$$\frac{\tau_a}{\tau_b} = \frac{\omega_a}{\omega_b} \quad (3.9)$$

$$\frac{\tau_a}{\tau_b} = - - - \quad (3.10)$$

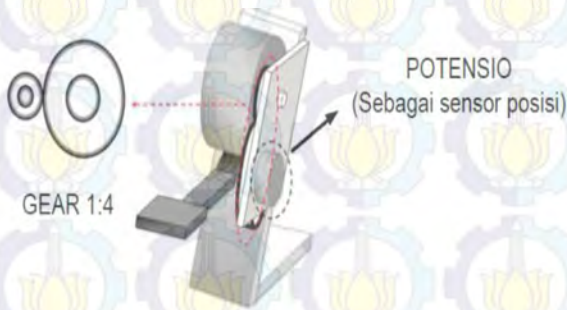
$$\tau_a = \tau_b \quad (3.11)$$



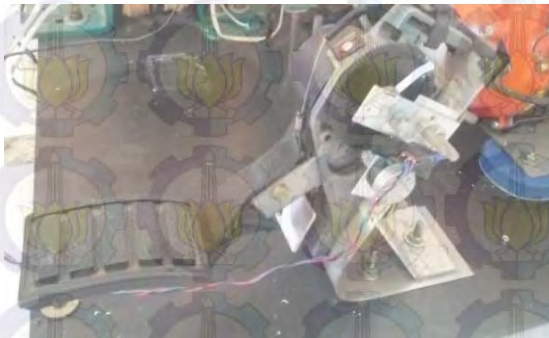
Dapat disimpulkan dari persamaan bahwa poros utama akan memiliki torsi sebesar empat kali torsi yang dihasilkan ICE.

#### 3.2.1.4 Sensor Posisi

Pada Simulator HEV, *set point* diberikan melalui pedal gas yang merupakan aktuator untuk mengatur lebar bukaan *throttle* pada karburator yang terhubung langsung dengan klep karburator melalui kabel. Pijakan pedal gas sebanding dengan bukaan *throttle* karburator. Oleh karena itu, untuk mengetahui bukaan *throttle* pada karburator dapat dilihat melalui pemberian sensor posisi berupa potensiometer pada pedal gas. Karena penempatan sensor posisi tidak dimungkinkan berada pada karburator, maka sensor posisi diletakkan pada pedal. Oleh karena itu derajat putaran pedal gas merepresentasikan bukaan klep pada karburator. Bentuk fisik sensor posisi pada HEV dapat dilihat pada **Gambar 3.8**.



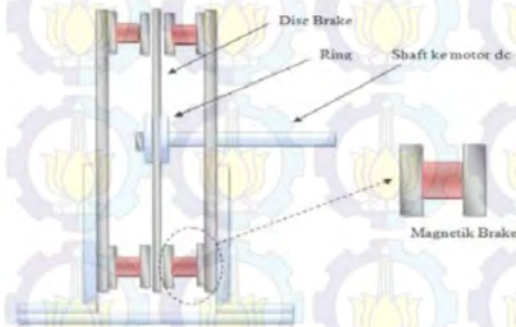
**Gambar 3.7** Ilustrasi Sensor Posisi pada Pedal Gas



**Gambar 3.8** Potensiometer pada Simulator HEV

### 3.2.1.5 Rem Elektromagnetik

Rem elektromagnetik pada simulator HEV ini berfungsi sebagai beban. Rem elektromagnetik diletakkan pada poros motor DC. Rem elektromagnet ini terdiri dari dua piringan besi yang diapit oleh dua penghasil medan magnet yang terdiri dari beberapa magnetik *brake* di sisi kiri dan kanan piringan. Magnetik *brake* ini terbuat dari konduktor yang telah dililit kumparan, karena dialiri arus listrik, maka akan timbul medan magnet. Konfigurasi dari rem elektromagnet ini dapat dilihat pada **Gambar 3.9**.



**Gambar 3.9** Konfigurasi Rem Elektromagnetik

Prinsip kerja dari rem elektromagnetik ini adalah ketika kecepatan poros utama tinggi, medan magnet yang dihasilkan oleh magnetik *brake* akan menimbulkan arus *eddy*, sehingga menyebabkan gaya lawan yang berlawanan dengan arah putar dari piringan besi. Hal ini akan menyebabkan terjadinya penurunan kecepatan putar pada motor DC. Gaya lawan inilah yang disebut sebagai beban bagi simulator HEV. Bentuk fisik rem elektromagnetik dapat dilihat pada **Gambar 3.10**.



**Gambar 3.10** Bentuk Fisik Rem Elektromagnetik



### 3.2.2 Perancangan Perangkat Elektronik

Perancangan elektronik dibutuhkan pada simulator HEV berupa rangkaian elektronik untuk mendukung sistem seperti Arduino, *rectifier*, dan *rotary encoder*.

#### 3.2.2.1 Sistem Minimum Arduino

Untuk keperluan akuisisi data dan pengolahan data pada Tugas Akhir ini digunakan sistem minimum Arduino R3 berbasis ATmega 328. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 analog *input*, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Pengiriman data dari sistem minimum Arduino ke PC/*laptop* dilakukan melalui USB. Bahasa yang digunakan oleh perangkat Arduino ini yaitu menggunakan bahasa C++ untuk pemrogramannya. *Software* Arduino ini dilengkapi dengan kumpulan *library* yang cukup lengkap, sehingga dapat membantu pengguna dalam penggunaannya. Sistem minimum Arduino Uno R3 secara lengkap dapat dilihat pada **Gambar 3.11**.




**Gambar 3.11** Bentuk Fisik Sistem Minimum Arduino Uno R3

#### 3.2.2.2 Driver Motor DC

*Driver* motor DC merupakan rangkaian yang digunakan untuk memberikan tegangan *input* pada motor DC. *Driver* berfungsi sebagai aktuator yang mengubah besaran sinyal kontrol menjadi besaran yang digunakan untuk mengoperasikan motor DC tersebut. Rangkaian *driver* motor DC memanfaatkan sinyal PWM yang dihasilkan oleh IC regulator dan memanfaatkan cara kerja dua transistor dimana arus yang mengalir

Bentuk fisik dari rangkaian *driver* motor DC dapat ditunjukkan pada Gambar 3.13.

**Gambar 3.12** Rangkaian *Driver Motor DC*



**Gambar 3.13** Bentuk Fisik *Driver Motor DC*

**3.2.2.3 Rangkaian Penyearah Gelombang Penuh**

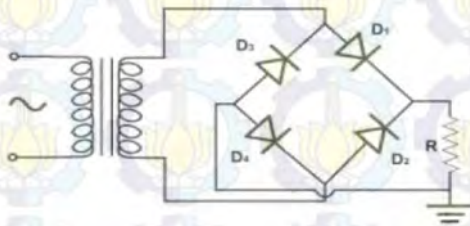
Motor DC membutuhkan suplai tegangan DC sebesar 190 V yang didapat dari tegangan jala-jala PLN sebesar 220 VAC 50 Hz yang merupakan tegangan AC kemudian disearahkan menjadi tegangan DC dengan rangkaian penyearah gelombang penuh. Rangkaian penyearah

### 3.2.2.3 Rangkaian Penyearah Gelombang Penuh

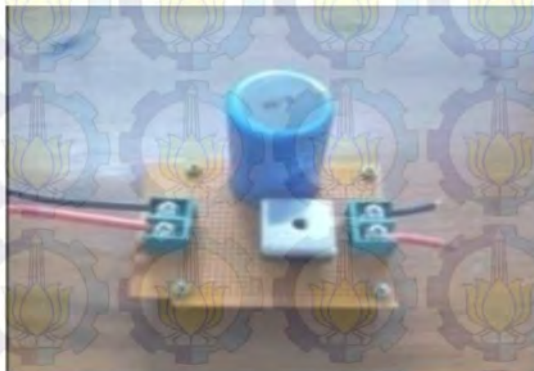
Motor DC membutuhkan suplai tegangan DC sebesar 190 V yang didapat dari tegangan jala-jala PLN sebesar 220 VAC 50 Hz yang merupakan tegangan AC kemudian disearahkan menjadi tegangan DC dengan rangkaian penyearah gelombang penuh. Rangkaian penyearah

gelombang ini memiliki komponen utama yaitu dioda. Dioda memiliki karakteristik *short circuit* ketika *forward bias* dan *open circuit* ketika *reverse bias*. Skema rangkaian penyearah gelombang penuh dapat dilihat pada **Gambar 3.14**. Bentuk fisik rangkaian penyearah gelombang penuh dapat dilihat pada **Gambar 3.15**.

Ketika tegangan masukan bernilai positif, maka dioda D1 dan D4 berada pada posisi *forward bias* sehingga tegangan keluaran pada R bernilai positif. Ketika tegangan masukan bernilai negatif, maka dioda D2 dan D3 berada pada posisi *forward bias* sehingga tegangan keluaran pada R akan bernilai positif. Hasil dari proses ini tidak didapatkan tegangan DC murni melainkan tegangan DC yang berubah secara periodik dan masih mengandung komponen tegangan bolak – balik. Variasi tegangan ini disebut dengan tegangan riak, sehingga untuk memperkecil riak tegangan dapat menggunakan filter kapasitor.



**Gambar 3.14** Skema Rangkaian Penyearah Gelombang Penuh



**Gambar 3.15** Bentuk Fisik Rangkaian Penyearah Gelombang Penuh pada Simulator HEV



### 3.2.2.4 Rotary Encoder

*Rotary encoder* digunakan sebagai sensor kecepatan pada HEV. Pada *rotary encoder* terdapat *optocoupler* ini dipasang pada *shaft* motor DC seperti pada **Gambar 3.16**, sehingga kecepatan motor DC bisa terbaca. *Rotary encoder* akan mendeteksi tiap lubang pada *encoder disk* dan mengubahnya dalam bentuk pulsa digital 0 1. *Rotary encoder* mendeteksi lubang pada *encoder disk* dengan menggunakan *optocoupler* yang mentransmisikan cahaya diantara celah *disk*, bila cahaya melewati lubang *disk* dan diterima *receiver*, maka sensor akan mengeluarkan pulsa 1, sedangkan saat cahaya tidak melewati lubang *disk*, maka sensor akan mengeluarkan pulsa 0. Modul *optocoupler* dan *encoder disk* dapat dilihat pada **Gambar 3.17**.

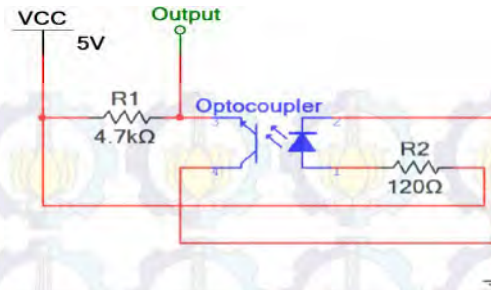


**Gambar 3.16** *Rotary Encoder* pada Simulator HEV



**Gambar 3.17** Modul *Optocoupler* dan *Encoder Disk*

Rangkaian *optocoupler* dibuat agar *optocoupler* bisa bekerja dengan baik dalam pengambilan data kecepatan motor DC. Pembuatan rangkaian tersebut berdasarkan data dari *datasheet* yang dimiliki *optocoupler*. Skema gambar rangkaian *optocoupler* ditunjukkan **Gambar 3.18**.



**Gambar 3.18** Skema Rangkaian *Optocoupler*

### 3.3 Perancangan Perangkat Lunak

Perangkat lunak diperlukan dalam perancangan sistem sebagai *interface* antara *plant* dan komputer. Perangkat lunak yang digunakan dalam pengerjaan Tugas Akhir ini antara lain adalah penggunaan perangkat lunak Arduino, MATLAB, dan LabView. Perangkat lunak digunakan untuk pendukung penggunaan perangkat keras, pengambilan data identifikasi *plant*, dan simulasi kontroler pada *plant*.

#### 3.3.1 *Software* Arduino

Penggunaan sistem Arduino pada Tugas Akhir ini membutuhkan perancangan program untuk menjalankan fungsi dari minimum sistem tersebut. Pada Arduino, program dituliskan pada *software* Arduino yang telah disediakan pada *Arduino.cc*, lalu program dituliskan pada Arduino *board* melalui serial USB yang dihubungkan pada komputer. Dalam Tugas Akhir ini, program Arduino digunakan untuk membaca *input analog* dari *output rotary encoder* untuk mengetahui respon kecepatan putar HEV. Tampilan program Arduino yang dirancang dapat dilihat pada **Gambar 3.19**.

Pada program Arduino yang dibuat dilakukan linearisasi antara kecepatan motor DC hasil pembacaan *rotary encoder* dengan kecepatan motor DC hasil pembacaan *tachometer digital* yang ditunjukkan pada **Persamaan (3.12)**. Hal ini dilakukan agar data kecepatan yang diolah oleh LabView hanya memiliki sedikit perbedaan dengan data kecepatan asli.

dengan, 
$$f(x) \quad x \quad (3.12)$$

$f_x$  : kecepatan *output* Arduino  
 $x$  : kecepatan *output* rotary encoder

```

encoder
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(encoder, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    delay(1);
    if (Serial.read() == 'M')
    {
      while (Serial.available())
      {
        Serial.read();
      }

      unsigned long waktu = pulseIn(encoder, HIGH);
      float jeda = waktu;
      jeda = jeda / 1000000;
      float kecepatan = (1 / (jeda * 72)) * 60;
      float y = 229.9536927+(0.680455391*kecepatan);
      word rpm = y;
      Serial.print(rpm);
    }
  }
}

```

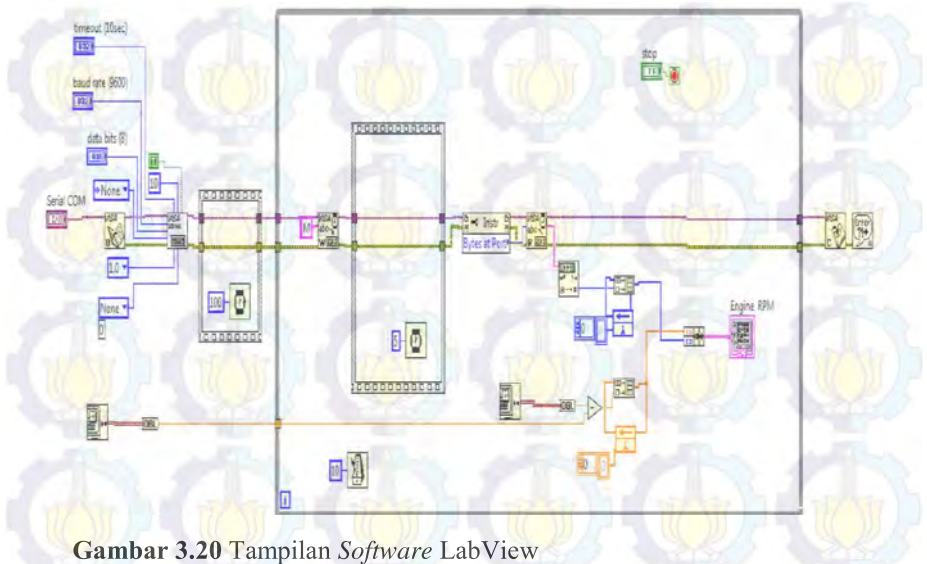
**Gambar 3.19** Tampilan Program Arduino

### 3.3.2 Software LabView

Perancangan dengan *software* LabView dilakukan untuk menampilkan data kecepatan motor DC dalam bentuk grafik. Selain itu



dengan LabView grafik data kecepatan juga dapat disimpan. Tampilan *software* LabView yang digunakan dapat dilihat pada **Gambar 3.20**.



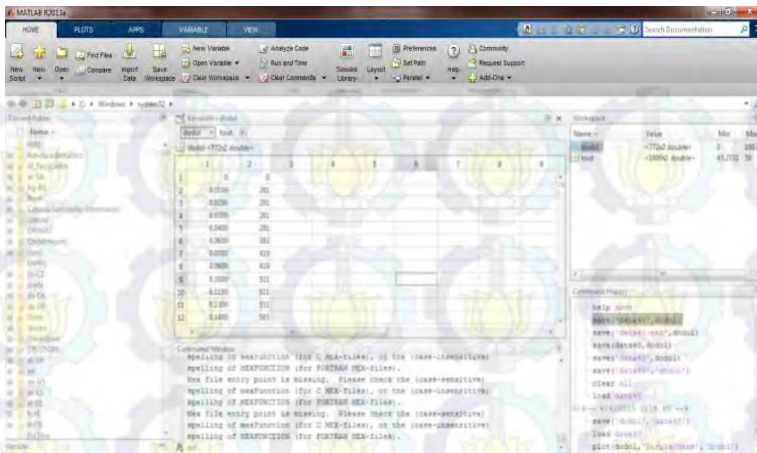
**Gambar 3.20** Tampilan *Software* LabView

### 3.3.3 *Software* MATLAB

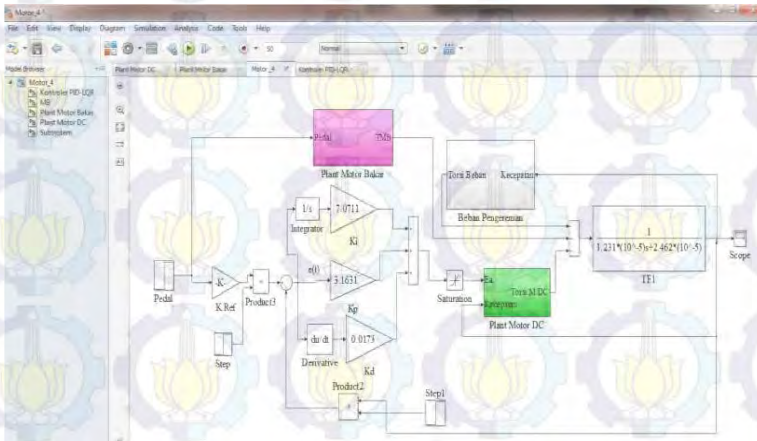
*Software* MATLAB yang digunakan pada Tugas Akhir ini adalah MATLAB R2013a. *Software* MATLAB digunakan untuk merancang simulasi sistem menggunakan menu Editor dan Simulink. MATLAB juga berfungsi sebagai *interface* antara mesin dan manusia atau *Human Machine Interface* (HMI) pada proses pengambilan data dan pengiriman data untuk ditampilkan dalam suatu grafik. Tampilan MATLAB R2013a dapat dilihat pada **Gambar 3.21** dan tampilan Simulink dari MATLAB dapat dilihat pada **Gambar 3.22**.

Pada Tugas Akhir ini, menu Editor digunakan untuk merancang program LQR untuk mendapatkan nilai  $K_p$ ,  $K_i$  dan  $K_d$  dengan memasukkan matriks **A**, matriks **B**, dan matriks pembobot sedangkan menu Simulink digunakan untuk merancang diagram blok keseluruhan sistem, termasuk di dalamnya untuk merancang kontroler PID yang nilainya telah di-*tuning* melalui program LQR.





Gambar 3.21 Tampilan MATLAB R2013a



Gambar 3.22 Tampilan Simulink pada MATLAB R2013a

### 3.4 Pengujian Sistem

Untuk mengetahui performansi pada *plant*, karakteristik dan kelayakan dari sistem tersebut, maka dibutuhkan pengujian pada beberapa sistem pada simulator HEV.

### 3.4.1 Pengujian Sensor

#### 3.4.1.1 Pengujian Rotary Encoder

Pada Tugas Akhir ini untuk mengukur kecepatan putar motor DC digunakan sensor *rotary encoder*. Pengujian sensor *rotary encoder* dilakukan dengan membandingkan hasil nilai pembacaan kecepatan putar *rotary encoder* dengan nilai pembacaan kecepatan putar yang diukur menggunakan dengan *tachometer digital*. Tujuannya adalah untuk mengetahui keakurasian dari pembacaan kecepatan putar. Bentuk fisik *tachometer digital* dapat dilihat pada **Gambar 3.23**.



**Gambar 3.23** Bentuk Fisik *Tachometer Digital*

Berdasarkan hasil pengujian didapatkan hasil pembacaan keluaran *rotary encoder* dengan kecepatan putar pada poros motor DC menggunakan *tachometer digital*. Data hasil pembacaan terhadap sensor *rotary encoder* dan *tachometer digital* ditunjukkan pada **Tabel 3.3**. Dari hasil pengujian tersebut dapat diketahui perbandingan perubahan besar kecepatan putar motor DC dan kecepatan putar *rotary encoder* dengan mencari linearisasi menggunakan regresi linear.

Persamaan linearisasi antara kecepatan motor DC hasil pembacaan *rotary encoder* dengan kecepatan motor DC hasil pembacaan *tachometer digital* yang ditunjukkan pada **Persamaan (3.13)** akan dimasukkan pada program Arduino. Hal ini dilakukan agar data kecepatan yang diolah oleh LabView hanya memiliki sedikit perbedaan dengan data kecepatan asli. Perbandingan data kecepatan hasil linearisasi dengan kecepatan asli dapat dilihat pada **Tabel 3.4**.

**Tabel 3.3** Data Hasil Pembacaan terhadap Sensor *Rotary Encoder* dan *Tachometer Digital*

No	Vin (V)	Kecepatan <i>Tachometer Digital</i> (rpm)	Kecepatan <i>Rotary Encoder</i> (rpm)
1	25	595	570
2	50	981	1070
3	80	1136	1320
4	90	1179	1400
5	100	1226	1550
6	120	1339	1602
7	150	1539	1881
8	160	1606	1911
9	170	1676	2164
10	180	1743	2209
11	190	1806	2314

Hasil linierisasi menggunakan regresi linear dapat dilihat pada **Persamaan (3.13)**.

$$f(x) \quad x \quad (3.13)$$

dengan,

$f$  : kecepatan *output* arduino (rpm)

$x$  : kecepatan *output* *rotary encoder* (rpm)

**Tabel 3.4** Perbandingan Data Kecepatan Hasil Linearisasi dengan Kecepatan Asli (dalam rpm)

Kecepatan untuk LabView (Hasil Linearisasi)	Kecepatan Asli ( <i>Tachometer Digital</i> )
592,3796066	595
981,3256628	981
1133,129503	1136



Kecepatan untuk Labview (Hasil Linearisasi)	Kecepatan Asli (Tachometer Digital)
1179,901497	1179
1229,955737	1226
1348,937125	1339
1531,922295	1539
1614,798986	1606
1677,982207	1676
1716,548588	1743
1819,118750	1806

#### 3.4.1.2 Pengujian Sensor Potensiometer

Masukan dari sistem HEV direpresentasikan dalam bukaan pedal. Sensor potensiometer dipasang untuk mengetahui derajat bukaan *throttle* pada karburator. Dengan adanya potensiometer dapat direpresentasikan perubahan hambatan pada potensiometer berbanding lurus dengan perubahan presentase bukaan *throttle* ICE. Hubungan antar hambatan potensiometer dengan presentase bukaan *throttle* berdasarkan **Persamaan (3.14)**.

$$f(x) = \frac{x - \dots}{\dots} \quad (3.14)$$

dengan,

$f(x)$  : presentase bukaan *throttle* (%)  
 $x$  : resistansi potensiometer (Ohm)

### 3.5 Proses Identifikasi dan Pemodelan Sistem

Untuk melakukan pengaturan terhadap suatu sistem, hal pertama yang harus diketahui adalah karakteristik dan hubungan *input-output* dari sistem tersebut. Hal ini dapat dilakukan dengan melakukan identifikasi pada sistem. Identifikasi dan pemodelan yang dilakukan pada Tugas Akhir ini terbagi menjadi 2 bagian, yaitu motor DC, dan ICE. Identifikasi untuk motor DC dilakukan dengan menggunakan identifikasi fisik dari konfigurasi motor DC paralel. Sedangkan untuk

identifikasi ICE dilakukan dengan sinyal uji *step* menggunakan metode Strejc.

### 3.5.1 Identifikasi dan Pemodelan Motor DC

#### 3.5.1.1 Identifikasi Parameter Motor DC

Identifikasi parameter motor DC dilakukan untuk mendapatkan parameter-parameter dari sistem berdasarkan hasil pengukuran *input-output plant*. Parameter-parameter yang diperoleh akan digunakan untuk mendapatkan model matematika dari motor DC.

Untuk keperluan pemodelan motor DC dibutuhkan beberapa parameter, yaitu konstanta gaya gerak listrik balik motor DC ( $K_{ggl}$ ), konstanta torsi motor DC  $K_{TMDC}$ , induktansi kumparan jangkar ( $L_a$ ) dan resistansi kumparan jangkar  $R_a$ , induktansi kumparan medan  $L_f$  dan resistansi kumparan medan ( $R_f$ ), dan parameter beban ( $J_m$ ,  $B_m$  dan  $B_l$ ).

#### a. Parameter Konstanta GGL Balik Motor DC

Percobaan untuk mendapatkan parameter konstanta balik motor DC dilakukan dengan memutar motor DC dengan motor DC hanya diberi masukan pada kumparan medannya saja

Percobaan ini dilakukan pada 5 kondisi tegangan masukan ke kumparan medan, yaitu 150 V, 160 V, 170 V, 180 V dan 190 V. Perubahan tegangan masukan ini dilakukan pada 3 kondisi kecepatan yang berbeda, yaitu 1000 rpm, 910 rpm dan 830 rpm. Setelah semua data dari percobaan didapatkan, data dimasukkan kedalam **Persamaan (3.15)** untuk mendapatkan nilai konstanta balik motor DC ( $K_{ggl}$ ). Dari data hasil percobaan pada **Tabel 3.5** didapatkan nilai rata-rata konstanta balik motor DC ( $K_{ggl}$ ) sebesar 0,71546.

$$K_{ggl} = \frac{E_a}{I_f \omega} \quad (3.15)$$

dengan,

$K_{ggl}$  : Konstanta gaya gerak listrik balik motor DC (V-sec/rad)

$E_a$  : Tegangan kumparan jangkar (V)

$I_f$  : Arus medan (A)

$\omega$  : Kecepatan putar (rpm)

**Tabel 3.5** Data Percobaan Konstanta Balik Motor DC

No	$V_{in}$ V	Kecepatan (rpm)	$I_f$ A	$E_a$ V	$K_{ggl}$
1	150	1000	0,0729	68	0,9327
2	160	1000	0,0795	70	0,8805
3	170	1000	0,0835	72	0,8622
4	180	1000	0,0945	73	0,7724
5	190	1000	0,1006	75	0,7455
6	150	910	0,0778	54	0,6512
7	160	910	0,0816	59	0,6550
8	170	910	0,0866	59	0,6812
9	180	910	0,0916	60	0,7230
10	190	910	0,0952	62	0,6940
11	150	830	0,0767	49	0,6388
12	160	830	0,0822	52	0,6236
13	170	830	0,0864	56	0,6481
14	180	830	0,0924	57	0,6168
15	190	830	0,0972	59	0,6069
Rata-rata $K_{ggl}$					0.71546

**b. Parameter Konstanta Torsi Motor DC**

Untuk mendapatkan parameter konstanta motor DC dibutuhkan beberapa parameter yaitu arus jangkar ( $I_a$ ), arus medan ( $I_f$ ), dan perubahan torsi motor DC ( $\Delta\tau_{mdc}$ ). Konstanta torsi motor DC  $K_{TMDC}$  diperoleh berdasarkan **Persamaan (3.16)**.

$$K_{TMDC} = \frac{\Delta\tau_{mdc}}{(I_{a \max} - I_{f \max}) - (I_{a \text{ nom}} - I_{f \text{ nom}})} \quad (3.16)$$

dengan,



- $K_{TMDC}$  : Konstanta torsi motor DC (V-sec/rad)  
 $\Delta\tau_{mdc}$  : Perubahan torsi motor DC (N-m)  
 $I_{a\ max}$  : Arus kumparan jangkar kondisi maksimal (A)  
 $I_{f\ max}$  : Arus medan kondisi maksimal (A)  
 $I_{a\ nom}$  : Arus kumparan jangkar kondisi nominal (A)  
 $I_{f\ nom}$  : Arus medan kondisi nominal (A)

Percobaan untuk mendapatkan parameter konstanta motor DC dilakukan pada kondisi maksimal dan nominal. Pada percobaan kondisi maksimal, motor DC dihidupkan dengan sumber tegangan sebesar 190 V dan beban rem elektromagnetik sebesar 80 V. Percobaan kondisi nominal dilakukan dengan cara menghidupkan motor DC dengan sumber tegangan sebesar 150 V dan beban rem elektromagnetik sebesar 40 V. Dari hasil percobaan didapatkan nilai arus jangkar ( $I_a$ ), dan arus medan ( $I_f$ ) pada saat kondisi maksimal dan kondisi nominal. Data hasil percobaan dapat dilihat pada **Tabel 3.6**.

**Tabel 3.6** Data Percobaan Kondisi Maksimal dan Nominal

Parameter	Nilai (A)
$I_{a\ max}$	6,4200
$I_{a\ nom}$	4,2800
$I_{f\ max}$	0,0763
$I_{f\ nom}$	0,0631

Setelah didapatkan nilai dari parameter arus jangkar ( $I_a$ ), dan arus medan ( $I_f$ ), langkah selanjutnya adalah mendapatkan nilai dari perubahan torsi motor ( $\Delta\tau_{mdc}$ ).  $\Delta\tau_{mdc}$  dapat diperoleh berdasarkan **Persamaan (3.17)** sebagai berikut :

$$P_{in} = V_{in}(I_a + I_f) = P_{rugi-rugi} + \tau_m \omega \quad (3.17)$$

dengan,

- $P_{in}$  : Daya *input* motor DC (Watt)  
 $V_{in}$  : Tegangan *input* motor DC (V)



$I_a$  : Arus kumparan jangkar (A)  
 $I_f$  : Arus medan (A)  
 $P_{rugi-rugi}$  : Daya rugi-rugi motor DC (Watt)  
 $\tau_m$  : Torsi motor DC (N-m)  
 $\omega$  : Kecepatan putar motor DC (rpm)

Percobaan pada kondisi maksimal dan minimal dilakukan hingga motor DC memiliki kecepatan 1450 rpm. Karena percobaan dilakukan pada kondisi kecepatan yang sama, maka nilai  $P_{rugi-rugi}$  pada kondisi maksimal dan minimal juga sama, sehingga dapat dieliminasi. **Persamaan (3.17)** dapat diubah menjadi **Persamaan (3.18)**.

$$\omega(\tau_{m\ max} - \tau_{m\ nom}) = \frac{V_{in\ max}(I_{a\ max} - I_{f\ max}) - V_{in\ nom}(I_{a\ nom} - I_{f\ nom})}{I_{f\ nom}} \quad (3.18)$$

$$\tau_{mdc} = \frac{V_{in\ max}(I_{a\ max} - I_{f\ max}) - V_{in\ nom}(I_{a\ nom} - I_{f\ nom})}{\omega} \quad (3.19)$$

dengan,

$V_{in\ max}$  : tegangan *input* motor DC saat beban 80 V (V)  
 $I_{a\ max}$  : arus kumparan jangkar saat beban 80 V (A)  
 $I_{f\ max}$  : arus medan saat beban 80 V (A)  
 $V_{in\ nom}$  : tegangan *input* motor DC saat beban 40 V (V)  
 $I_{a\ nom}$  : arus kumparan jangkar saat beban 40 V (A)  
 $I_{f\ nom}$  : arus medan saat beban 40 V (V)

Data hasil pengukuran parameter arus jangkar, arus medan, dan tegangan input saat kondisi maksimal dan nominal dapat dilihat pada **Tabel 3.7**. Substitusi data hasil pengukuran pada **Tabel 3.7** ke dalam Persamaan (3.19).

$$\Delta\tau_{mdc} = \frac{V_{in\ max}(I_{a\ max} - I_{f\ max}) - V_{in\ nom}(I_{a\ nom} - I_{f\ nom})}{\omega}$$

$$\Delta\tau_{mdc} = \frac{(\quad) - (\quad)}{\quad}$$

$$\Delta\tau_{mdc} = 0.138$$

**Tabel 3.7** Data Hasil Pengukuran Parameter

Parameter	Nilai
$I_a \text{ max}$	6,4200 A
$I_a \text{ nom}$	4,2700A
$I_f \text{ max}$	0,0763 A
$I_f \text{ nom}$	0,0631 A
$V_{in \text{ max}}$	63,500 V
$V_{in \text{ nom}}$	49,000 V

Setelah mendapatkan nilai  $\Delta\tau_{mdc}$ , substitusikan nilai  $\Delta\tau_{mdc}$  dan nilai pada **Tabel 3.7** pada **Persamaan (3.16)** sehingga didapatkan nilai dari konstanta torsi motor DC ( $K_{TMDC}$ ).

$$K_{TMDC} = \frac{\Delta\tau_{mdc}}{(I_a \text{ max} - I_a \text{ nom}) - (I_f \text{ max} - I_f \text{ nom})}$$

$$K_{TMDC} = \frac{(\quad) - (\quad)}{(\quad) - (\quad)}$$

$$K_{TMDC} = 0.624$$

### c. Induktansi dan Resistansi

Untuk keperluan pemodelan motor DC, parameter yang dibutuhkan selanjutnya adalah nilai induktansi dan resistansi dari kumparan motor, baik kumparan jangkar maupun medan  $L_a$   $R_a$   $L_f$   $R_f$ . Pengukuran parameter ini dilakukan dengan menggunakan alat LCR meter. Hasil pengukuran parameter  $L_a$   $R_a$   $L_f$   $R_f$  dapat dilihat pada **Tabel 3.8**.

**Tabel 3.8** Hasil pengukuran parameter  $L_a$   $R_a$   $L_f$   $R_f$

Parameter	Nilai
$R_a$	19,748 $\Omega$
$R_f$	2377,120 $\Omega$
$L_a$	0,091160 H
$L_f$	23,34 H

#### d. Parameter Beban

Beban pada motor DC pada terbagi menjadi beban rem elektromagnetik dan beban komponen penghubung antara ICE dengan motor DC. Parameter-parameter yang dibutuhkan yaitu konstanta rem elektromagnetik ( $K_d$ ), momen inersia beban  $J_m$ , koefisien gesekan viskos beban  $B_m$ , dan beban rem elektromagnetik ( $B_l$ ).

Parameter konstanta rem elektromagnetik ( $K_d$ ), koefisien gesekan viskos beban  $B_m$  didapatkan berdasarkan **Persamaan (3.20)**.

$$\omega = \frac{B_m - \omega I_l}{K_d} \quad K_{TMDC} \quad I_a \quad I_f \quad (3.20)$$

dengan,

- $\omega$  : Kecepatan putar motor DC (rpm)
- $B_m$  : Koefisien gesek viskos beban (N-m/rad/s)
- $I_l$  : Arus rem elektromagnetik (A)
- $K_d$  : Konstanta rem elektromagnetik
- $K_{TMDC}$  : Konstanta torsi motor DC (N-m/A)
- $I_a$  : Arus kumpar jangkar (A)
- $I_f$  : Arus medan (A)

Percobaan untuk mendapatkan parameter beban juga dilakukan pada kondisi maksimal (beban 80 V) dan kondisi nominal (beban 40 V), namun untuk mendapatkan parameter beban harus diketahui arus rem elektromagnetik. Berikut ini merupakan data hasil pengukuran arus rem elektromagnetik pada **Tabel 3.9**.

**Tabel 3.9** Data Hasil Pengukuran Arus Rem Elektromagnetik

Beban (V)	$I_l$ (A)	$\Omega$ (rpm)
40	0,56	1450
80	1,14	1450

Parameter pada **Tabel 3.9** disubstitusikan ke dalam **Persamaan (3.20)**, sehingga didapatkan **Persamaan (3.21)** dan **Persamaan (3.22)**.

saat beban 80 V ( $I_l$ ), sehingga didapatkan :

$$\begin{aligned} B_m - \\ B_m - \end{aligned} K_d \quad K_d \quad (3.21)$$

saat beban 40 V ( $I_l$ ), sehingga didapatkan :

$$\begin{aligned} B_m - \\ B_m - \end{aligned} K_d \quad K_d \quad (3.22)$$

eliminasi **Persamaan (3.21)** dan **Persamaan (3.22)**, sehingga didapatkan nilai  $K_d = -1.636 \cdot 10^{-4}$  dan  $B_m = 2.462 \cdot 10^{-5}$ .

Koefisien gesek viskom rem elektromagnetik diperoleh berdasarkan **Persamaan (3.23)** sebagai berikut :

$$B_l = -K_d I_l \quad (3.23)$$

Beban yang digunakan pada Tugas Akhir ini yaitu pada saat kondisi nominal, dimana arus rem elektromagnetik  $I_l$  sebesar 0,56 A. Dengan mensubstitusikan arus rem dan nilai  $K_d$  yang telah didapatkan ke **Persamaan (3.23)**, diperoleh :

$$\begin{aligned} B_l &= -(-) \\ B_l &= -(-) \\ B_l &= \text{N-m/rad/sec} \end{aligned}$$

Parameter momen inersia beban ( $J_m$ ) didapatkan dari respon sistem motor DC dengan mengambil data respon kecepatan dari motor DC saat diberikan *input* tegangan berbentuk sinyal *step* seperti pada **Gambar 3.24**. Sinyal *step* disini adalah tegangan dengan perubahan dari 0 V sampai 150 V dalam waktu yang singkat. *Transfer function* beban dapat dilihat berdasarkan **Persamaan (3.24)** :

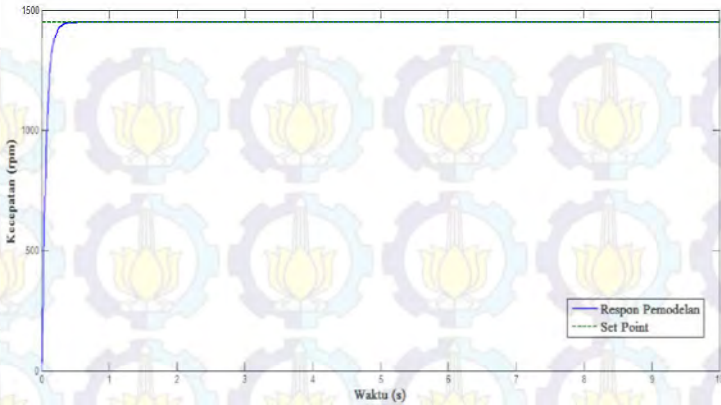
$$\frac{J_m S}{B_m} = \frac{J_m / B_m}{S} = \frac{\tau S}{\tau S} \quad (3.24)$$

sehingga,

$$\tau = \frac{J_m}{B_m} \quad (3.25)$$



$$J_m \quad \tau \quad B_m \quad (3.26)$$

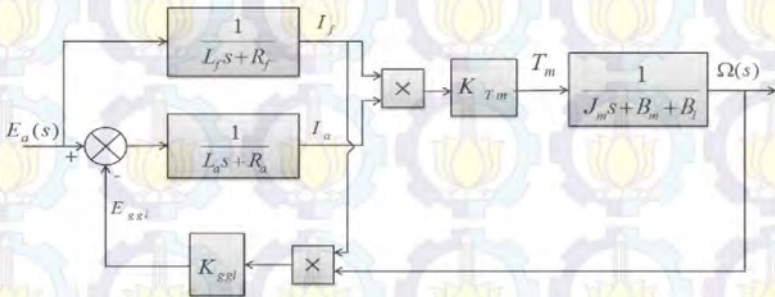


**Gambar 3.24** Respon Motor DC saat Diberikan Input Sinyal Step

Berdasarkan respon pada **Gambar 3.24** didapatkan nilai  $\tau = 0.5$ . Diketahui nilai  $B_m = 2.462 \cdot 10^{-5}$  N-m/rad/s, sehingga nilai  $J_m$  dapat diperoleh dengan mensubstitusi kedua nilai tersebut pada **Persamaan (3.26)**.

$$\frac{J_m}{J_m} = \text{kg-m}$$

. Parameter motor DC yang telah didapatkan dimasukkan ke dalam diagram blok motor DC seperti yang ditunjukkan yang ditunjukkan **Gambar 3.25**.

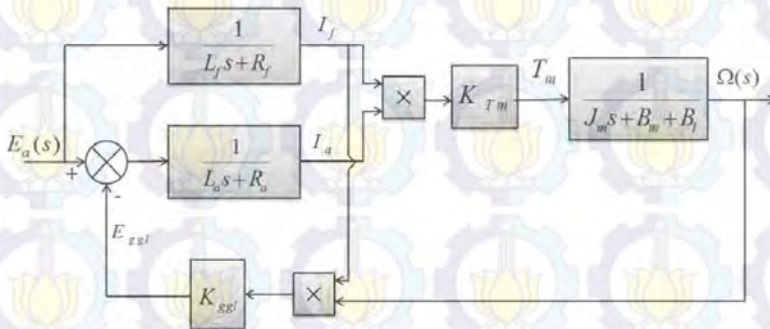


**Gambar 3.25** Diagram Blok Motor DC

Variasi beban yang diperbolehkan dalam pengujian kontroler harus sesuai dengan pemodelan nominal untuk ICE (d disesuaikan dengan *range* pembebanan nominal), sehingga dibatasi beban yang diperbolehkan adalah pada *range* arus beban 0,56 A– 1,14 A.

### 3.5.1.2 Pemodelan Motor DC

Pada Tugas Akhir ini digunakan Motor DC dengan Konfigurasi Paralel. Motor DC dengan konfigurasi paralel memiliki karakteristik non linear terhadap arus jangkar, arus medan, dan torsi motor. Diagram blok fisik motor DC konfigurasi paralel dapat dilihat pada **Gambar 3.26**.



**Gambar 3.26** Diagram Blok Fisik Motor DC Konfigurasi Paralel

Karena terdapat beberapa parameter pada konfigurasi motor DC paralel memiliki karakteristik non linier, untuk keperluan perancangan kontroler yang memerlukan model linear maka beberapa persamaan pada motor DC harus dilinearisasi yaitu torsi motor ( $T_m$ ) dan konstanta balik motor ( $E_{ggl}$ )

#### a. Linearisasi Persamaan Torsi Motor

$$T_m = K_{TM} I_a I_f \quad (3.27)$$

$$\tilde{T}_m = \left. \frac{\partial K_{TM} I_a I_f}{\partial I_a} \right|_{i_a = i_{aop}} \tilde{i}_a + \left. \frac{\partial K_{TM} I_a I_f}{\partial I_f} \right|_{i_f = i_{fop}} \tilde{i}_f$$

$$\tilde{T}_m = K_{TM} I_{fop} \tilde{I}_a + K_{TM} I_{aop} \tilde{I}_f \quad (3.28)$$

#### b. Linearisasi Persamaan $E_{ggl}$

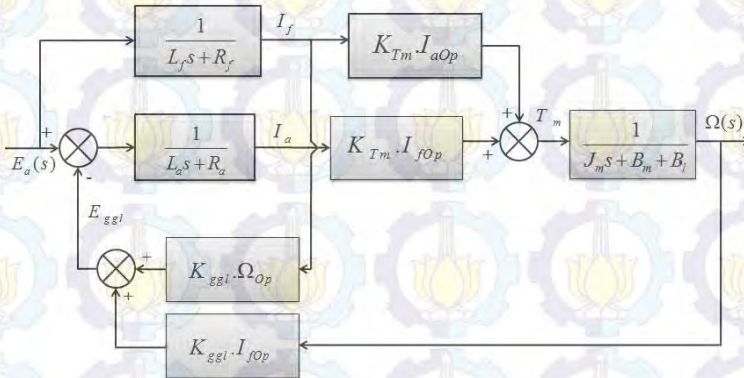
$$E_{ggl} = K_{ggl} I_f \Omega \quad (3.29)$$

$$\begin{aligned} \tilde{E}_{ggl} &= \left. \frac{\partial K_{ggl} I_f \Omega}{\partial I_f} \right|_{\Omega = \Omega_{op}} \tilde{I}_f + \left. \frac{\partial K_{ggl} I_f \Omega}{\partial \Omega} \right|_{I_f = I_{fop}} \tilde{\Omega} \\ \tilde{E}_{ggl} &= K_{ggl} I_f \Omega_{op} \tilde{I}_f + K_{ggl} I_{fop} \Omega \tilde{\Omega} \end{aligned} \quad (3.30)$$

dengan,

- $T_m$  : Torsi motor DC (N-m)
- $K_{TM}$  : Konstanta torsi motor (N-m/A)
- $L_a$  : Induktansi kumparan jangkar (H)
- $R_a$  : Reaktansi kumparan jangkar ( $\Omega$ )
- $L_f$  : Induktansi medan (H)
- $R_f$  : Reaktansi medan ( $\Omega$ )
- $E_{ggl}$  : Gaya gerak listrik balik (V)
- $K_{ggl}$  : Konstanta gaya gerak listrik balik (V-sec/rad)
- $I_{aop}$  : Arus kumparan jangkar beban nominal (A)
- $I_{fop}$  : Arus medan saat beban nominal (A)
- $\Omega_{op}$  : Kecepatan putar saat beban nominal (rpm)

Setelah beberapa persamaan yang non linear dilinearisasi, maka **Gambar 3.26** dapat diubah menjadi **Gambar 3.27**.



**Gambar 3.27** Diagram Blok Fisik Motor DC Konfigurasi Paralel Hasil Linearisasi



Dari **Gambar 3.27** dapat dicari model matematika motor DC melalui beberapa persamaan :

$$T_m = \frac{K_{TM} I_{fop}}{L_a s} (E_a - E_{ggl}) - \frac{K_{TM} I_{aop}}{L_f s} E_a \quad (3.31)$$

$$E_{ggl} = \frac{K_{ggl} \Omega_{op}}{L_f s} E_a - K_{ggl} I_{fop} \Omega \quad (3.32)$$

substitusi **Persamaan (3.32)** ke **Persaman (3.31)**, sehingga didapat **Persamaan (3.33)** sebagai berikut :

$$\begin{aligned} T_m &= \frac{K_{TM} I_{fop}}{L_a s} \left( E_a - \frac{K_{ggl} \Omega_{op}}{L_f s} E_a - K_{ggl} I_{fop} \Omega \right) - \frac{K_{TM} I_{aop}}{L_f s} E_a \\ T_m &= \frac{K_{TM} I_{aop}}{L_f s} E_a - \frac{K_{TM} I_{fop}}{L_a s} \left( E_a - \frac{K_{ggl} \Omega_{op}}{L_f s} E_a - K_{ggl} I_{fop} \Omega \right) \\ T_m &= E_a \left[ \frac{K_{TM} I_{aop}}{L_f s} - \frac{K_{TM} I_{fop}}{L_a s} + \frac{K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s) L_a s} \right] - \underbrace{\frac{K_{ggl} I_{fop} \Omega K_{TM} I_{fop}}{L_a s}}_{\text{disturbance}} \\ \frac{T_m}{E_a} &= \frac{K_{TM} I_{aop}}{L_f s} - \frac{K_{TM} I_{fop}}{L_a s} + \frac{K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s) L_a s} \\ \frac{T_m}{E_a} &= \frac{K_{TM} I_{aop} (L_a s - R_a) - K_{TM} I_{fop} (L_f s - R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s - R_f) L_a s - R_a} \quad (3.33) \end{aligned}$$

Masukkan nilai parameter motor DC yang telah didapatkan pada perhitungan sebelumnya ke dalam **Persamaan (3.33)**, sehingga didapatkan hubungan *input-output* dalam fungsi alih pada **Persamaan (3.34)** sebagai berikut :

$$\frac{T_m}{E_a} = \frac{50 \ 0 \ 01s \ 1}{s^2 \ 321 \ 15s \ 22249 \ 9} \quad (3.34)$$

Nilai  $\Omega$  dapat diperoleh dengan menggunakan **Persamaan (3.35)** :

$$\Omega = T_m \frac{1}{J_m s^2 + B_m s + B_l} \quad (3.35)$$

$$\Omega \left[ \frac{K_{TM} I_{aop} (L_a s \ R_a) \ K_{TM} I_{fop} (L_f s \ R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s \ R_f) \ L_a s \ R_a} \right] E_a \frac{1}{J_m s \ B_m \ B_l}$$

$$\frac{\Omega}{E_a} = \frac{K_{TM} I_{aop} (L_a s \ R_a) \ K_{TM} I_{fop} (L_f s \ R_f) - K_{ggl} \Omega_{op} K_{TM} I_{fop}}{(L_f s \ R_f) (L_a s \ R_a) J_m s \ B_T} \quad (3.36)$$

dengan,

- $T_m$  : Torsi motor (N-m)  
 $L_a$  : Induktansi kumparan jangkar (H)  
 $R_a$  : Reaktansi kumparan jangkar ( $\Omega$ )  
 $L_f$  : Induktansi medan (H)  
 $R_f$  : Reaktansi medan ( $\Omega$ )  
 $K_{ggl}$  : Konstanta gaya gerak listrik balik (V-s/rad)  
 $I_{aop}$  : Arus kumparan jangkar saat beban nominal (A)  
 $I_{fop}$  : Arus medan saat beban nominal (A)  
 $\Omega_{op}$  : Kecepatan putar saat beban nominal (rpm)

Masukkan nilai parameter yang telah didapatkan pada perhitungan sebelumnya ke dalam **Persamaan (3.36)**, sehingga didapatkan hubungan *input-output* dalam fungsi alih pada **Persamaan (3.37)** sebagai berikut :

$$\frac{\Omega}{E_a} = \frac{s}{s \quad s \quad s} \quad (3.37)$$

karena nilai  $L_f \ L_a \ J_m$  sangat kecil maka dapat diabaikan, sehingga **Persamaan (3.37)** menjadi sebagai berikut:

$$\frac{\Omega}{E_a} = \frac{s}{s \quad s} \quad (3.38)$$

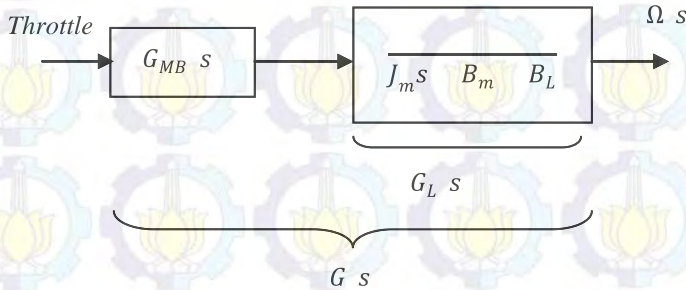
$$\frac{\Omega}{E_a} = \frac{s}{s \quad s} \quad (3.39)$$

Pada **Persamaan (3.39)** diatas terdapat *time delay* 0,01 detik. Karena nilainya kecil maka dapat diabaikan, sehingga **Persamaan (3.39)** dapat diubah menjadi sebagai berikut :

$$G(s) = \frac{\Omega}{E_a} \frac{12295.58}{s^2 + 76.4785s + 635.9768} \quad (3.40)$$

### 3.5.2 Identifikasi dan Pemodelan ICE

Untuk mendapatkan permodelan ICE, dilakukan identifikasi statis menggunakan metode Strejc. Proses pemodelan Strejc memanfaatkan grafik respon dari ICE saat diberikan masukan berupa sinyal *step*. Dari grafik respon akan didapatkan parameter yang akan menghasilkan model matematika yang mendekati respon ICE sebenarnya. Hasil pemodelan dengan metode Strejc adalah pendekatan dari pemodelan ICE yang telah terhubung dengan beban seperti yang dijelaskan pada **Gambar 3.28**.



**Gambar 3.28** Diagram Blok Pemodelan Strejc

sehingga didapatkan **Persamaan (3.41)**,

$$G_{MB}(s) = \frac{G s}{G_L s} \quad (3.41)$$

Pemodelan ICE terbagi menjadi 3, yaitu saat pembebanan minimal, nominal dan pembebanan maksimal.

#### a. Pembebanan Minimal

Pemodelan dengan pembebanan minimal dilakukan tanpa diberikan tambahan beban dari rem elektromagnetik. Tetapi untuk pengukuran kecepatan putar ICE membutuhkan penyatuan poros dengan motor DC yang telah ter-*couple* dengan *rotary encoder* sebagai pembacaan sensor, sehingga poros motor DC yang telah dihubungkan dengan ICE menggunakan gir dapat dianggap sebagai beban minimal

untuk ICE berbeban. Nilai  $y_{ss}$  merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan untuk ICE beban nominal dapat dilihat pada **Gambar 3.29**. Dari gambar tersebut kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$\frac{Y_{ss}}{X_{ss}} = K \frac{y_{ss}}{x_{ss}} = 36,97$$

Dari grafik respon pada **Gambar 3.29** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat kan nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$\frac{T_U}{T_U + T_N} = \frac{T_U}{T_N}$$

setelah itu dapat dicari nilai para meter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\tau = \frac{T_U}{2}$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$\frac{y_i}{y_{ss}} = 0,264$$

Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.



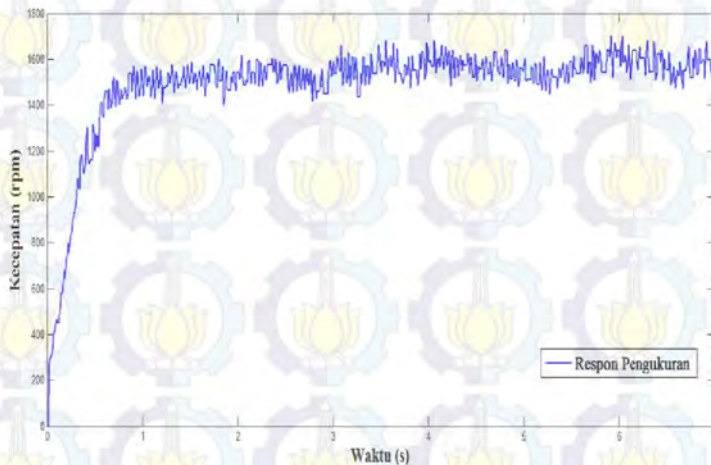
$$\tau_{ST} = \frac{t_i}{n} - \frac{\tau_{ST}}{n}$$

$$\tau_{ST}$$

sehingga didapat pendekatan model orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST} = \frac{K}{\tau_{ST} s}$$

$$G = \frac{36.97}{0.135s + 1^2}$$



**Gambar 3.29** Respon Kecepatan Putar ICE Beban Minimal

#### b. Pembebanan Nominal

Pembebanan nominal dilakukan dengan memberikan tambahan beban dari rem elektromagnetik. Setelah dilakukan beberapa kali percobaan, maka ditentukan bahwa beban nominal adalah beban dengan tegangan input pada rem elektromagnetik sebesar 40 V. Nilai merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan untuk ICE beban nominal dapat dilihat pada **Gambar 3.30**. Dari gambar tersebut

kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$K \frac{Y_{ss}}{X_{ss}} = 35,14$$

Dari grafik respon pada **Gambar 3.30** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat kan nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$\begin{matrix} T_U \\ T_U \\ T_N \end{matrix}$$

setelah itu dapat dicari nilai para meter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\tau = 0,0175$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$\begin{matrix} y_i \\ y_i \\ t_i \end{matrix}$$

Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.

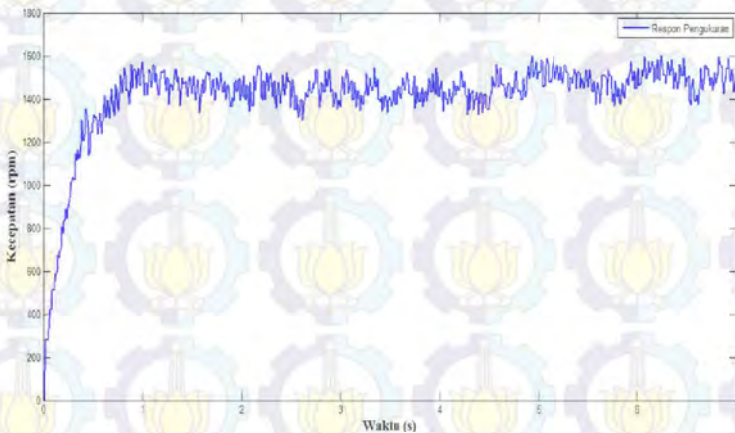
$$\tau_{ST} = \frac{t_i}{n-1}$$

$$\tau_{ST}$$

sehingga didapat pendekatan mode orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST} s = \frac{K}{\tau_{ST} s}$$

$$G s = \frac{35.14}{0.125s + 1}^2$$



**Gambar 3.30 Respon Kecepatan Putar ICE Beban Nominal**

### c. Metode Pembebanan Maksimal

Pembebanan maksimal dilakukan dengan memberi beban pada ICE berupa rem elektromagnetik. Setelah dilakukan beberapa percobaan, maka ditentukan bahwa beban nominal adalah beban dengan tegangan input pada rem elektromagnetik sebesar 80 V. Nilai merupakan nilai bukaan *throttle* yang didapatkan dengan menginjak pedal gas hingga kecepatan putar 1500 rpm. Hasil pemodelan ICE beban maksimal dapat dilihat pada **Gambar 3.31**. Dari gambar tersebut kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, seperti perhitungan berikut :

$$Y_{ss}$$



$$K \frac{Y_{ss}}{X_{ss}} = 32,11$$

Dari grafik respon pada **Gambar 3.31** kemudian ditarik garis singgung untuk mencari parameter respon menurut metode Strejc, maka didapat nilai  $T_U$  dan  $T_N$  sebagai berikut :

$$\begin{matrix} T_U \\ T_U \\ T_N \end{matrix}$$

setelah itu dapat dicari nilai parameter  $\tau$  berdasarkan **Persamaan (2.10)**.

$$\tau = 0,07$$

Berdasarkan nilai  $\tau$ , dapat dilakukan pendekatan model orde ke-n berdasarkan taksiran orde pada tabel dan dipatkan bahwa orde untuk pendekatan model adalah orde ke-2 dengan  $y_i$  yang merupakan titik singgung antara respon sistem dengan garis singgung. Setelah itu dapat dicari  $t_i$  saat respon sistem mencapai 0,264 dari  $y_{ss}$ .

$$\begin{matrix} y_i \\ y_i \\ t_i \end{matrix} \quad y_{ss}$$

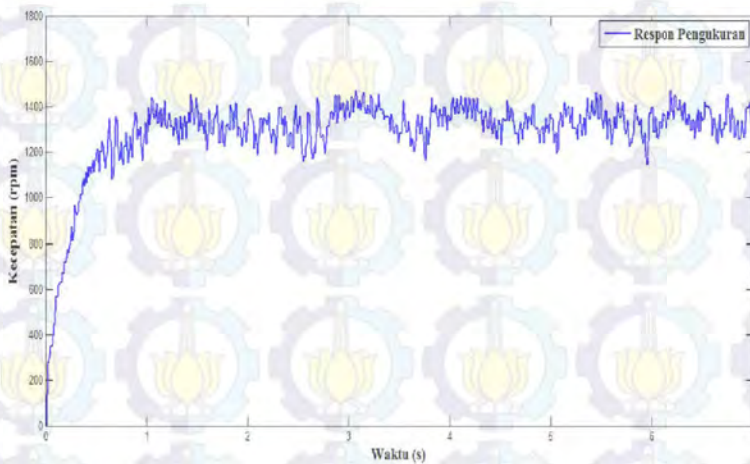
Waktu konstan  $\tau_{ST}$  dapat ditentukan berdasarkan **Persamaan (2.12)**.

$$\tau_{ST} = \frac{t_i}{n-1}$$

sehingga didapat pendekatan mode orde ke-2 berdasarkan **Persamaan (2.11)** sebagai berikut :

$$G_{ST} = \frac{K}{\tau_{ST} s}$$

$$G \ s \quad \frac{32 \ 11}{0 \ 130s \ 1 \ 2}$$



**Gambar 3.31** Respon Kecepatan Putar ICE Beban Maksimal

### 3.6 Pengujian dan Validasi

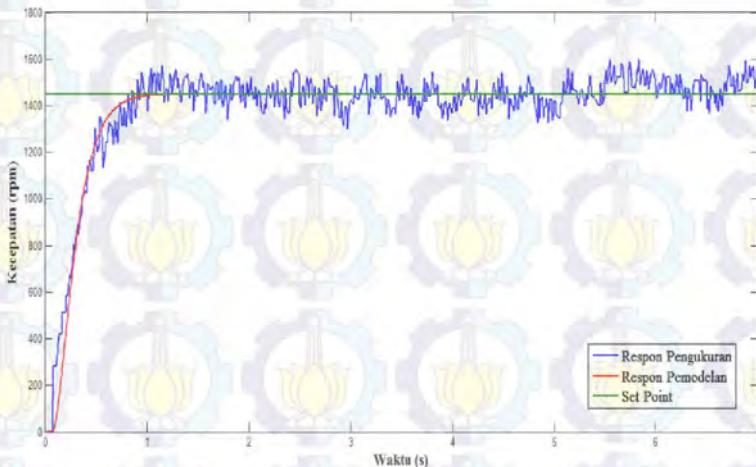
Setelah didapatkan beberapa model matematika pada *plant*, kemudian dilakukan pengujian dan validasi model matematika dengan cara membandingkan antara hasil perhitungan dengan hasil pengukuran respon. Hasil perbandingan tersebut akan divalidasi dengan menggunakan RMSE. Kemudian dihitung nilai *Relatif* RMSE (RRMSE) terhadap  $Y_{ss}$  dalam presentase. Pada kondisi sebenarnya, poros dari kedua penggerak ICE dan motor DC saling terhubung dari awal mula kerja sistem. Pada Tugas Akhir ini, karena pemodelan motor DC berbeban nominal sehingga pengujian dan validasi model serta pemodelan *plant* ICE yang dipilih adalah ICE berbeban nominal. Proses perhitungan identifikasi statis untuk mendapatkan pemodelan terhadap *plant* dilakukan sebanyak 5 kali kemudian dilakukan validasi pada model yang didapatkan.

Persamaan model dan validasi *plant* ICE beban nominal dapat dilihat pada **Tabel 3.10**, sedangkan **Gambar 3.32** merupakan hasil Identifikasi dan pemodelan ICE beban nominal.

**Tabel 3.10** Persamaan Model dan Validasi *Plant* ICE Beban Nominal

No	Persamaan Model <i>Plant</i>	RRMSE
1	$G(s) = \frac{1}{s}$	10,08 %
2	$G(s) = \frac{1}{s}$	10,04 %
3	$G(s) = \frac{1}{s}$	9,90 %
4	$G(s) = \frac{1}{s}$	10,14 %
5	$G(s) = \frac{1}{s}$	9,94 %

Hasil respon *plant* motor DC yang dipilih dengan menggunakan metode Strejc kemudian dibandingkan dengan metode identifikasi lainnya. Setelah melakukan beberapa pengujian dengan beberapa metode dapat dilihat bahwa RMSE terkecil dapat dicapai dengan menggunakan metode Strejc sehingga dapat disimpulkan bahwa hasil *transfer function* motor DC dengan menggunakan metode Strejc lebih mirip dengan respon asli. Hasil pengujian metode pendekatan model dapat dilihat pada **Tabel 3.11**.



**Gambar 3.32** Hasil Identifikasi dan Pemodelan ICE Beban Nominal

**Tabel 3.11** Pengujian Metode Pendekatan Model

No	Metode	Persamaan Model <i>Plant</i>	RRMSE
1	Latzel	$G \ s \ \underline{\hspace{2cm}}$	15,81 %
2	Viteckova 2 <sup>nd</sup> Order	$G \ s \ \underline{\hspace{2cm}}$	11,94 %
3	Strejc	$G \ s \ \underline{\hspace{2cm}}$	9,90 %

Sehingga didapatkan model matematika ICE berdasarkan **Persamaan (3.41)** :

$$G_{MB}(s) = \frac{G \ s}{G_L \ s}$$

$$G_{MB}(s) = \frac{s}{( \quad )s ( \quad ) \quad}$$

$$G_{MB}(s) = \frac{(43 \ 25734 \ 10^{-5})s \ 408 \ 453304 \ 10^{-5}}{0 \ 015625s^2 \ 0 \ 25s \ 1}$$

### 3.7 Perancangan Kontroler PID-LQR.

Model matematika motor DC dari proses identifikasi seperti ditunjukkan pada **Persamaan (3.40)**. Diketahui bahwa *plant* merupakan orde dua dengan input  $u(t)$ , output  $y(t)$ , dan fungsi alih *plant* sebagai berikut :

$$\frac{U \ s}{s} = \frac{a}{s \ \xi \omega_n s \ \omega_n} \frac{-E \ s}{U \ s} \quad (3.42)$$

$$\frac{U \ s}{s} = \frac{s}{s} \frac{-E \ s}{U \ s} \quad (3.43)$$

$$( \quad ) [s \quad s] - U(s)$$

$$s \ ( \quad ) \quad s \ ( \quad ) \quad ( \quad ) - U(s)$$

$$\ddot{e} \quad \dot{e} \quad e - u$$



$$\ddot{e} = \dot{e} = e = u \quad (3.44)$$

dimisalkan sebuah *state* baru,

$$x = \int e(t)dt \quad x = e(t) \quad x = \frac{de}{dt} \quad (3.45)$$

$$\dot{x} = e = x \quad (3.46)$$

$$\dot{x} = \dot{e} = x \quad (3.47)$$

$$\dot{x} = \ddot{e} \quad (3.47)$$

dengan,

- $x$  Augmented State
- $x$  Kecepatan putar motor DC
- $x$  Percepatan motor DC

Masukkan **Persamaan (3.44)** ke **Persamaan (3.47)**, sehingga didapatkan **Persamaan (3.48)** sebagai berikut :

$$\begin{aligned} \dot{x} &= \ddot{e} \\ \dot{x} &= e \\ \dot{x} &= x \\ \dot{x} &= x \end{aligned} \quad \begin{aligned} \dot{e} &= e \\ x &= x \\ x &= x \\ x &= x \end{aligned} \quad \begin{aligned} e &= u \\ x &= u \\ x &= u \\ x &= u \end{aligned} \quad (3.48)$$

representasi *plant* dalam *state space* dapat dibentuk sebagai berikut:

$$\begin{bmatrix} \dot{x} \\ \dot{x} \\ \dot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} + \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} u$$

sehingga diperoleh parameter *plant* sebagai berikut:

$$A = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \quad B = \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix}$$

Setelah diperoleh parameter *plant*, ditentukan matriks pembobot **Q** dan **R** melalui *trial and error*. Matriks **A**, matriks **B**, matriks pembobot

**Q** dan **R** digunakan untuk memperoleh matriks Riccati **P** dengan menyelesaikan persamaan *Algebraic Riccati Equation* (ARE) seperti pada **Persamaan (2.47)**. Pada Tugas Akhir ini dilakukan pemilihan matriks **Q** dan **R** sebanyak 4 kali dengan matriks **Q** semidefinit positif dan matriks **R** definit positif sehingga menghasilkan 4 nilai matriks Riccati **P** yang berbeda-beda. Nilai matriks Riccati **P** melalui penyelesaian ARE ditunjukkan pada **Tabel 3.12**.

**Tabel 3.12** Nilai Matriks **P** Melalui Penyelesaian ARE

Matriks Pembobot		Matriks P
R	Q	
$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$
$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$
$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$
$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$	$\begin{bmatrix} & \end{bmatrix}$

Melalui penyelesaian persamaan Kalman *gain* pada **Persamaan (2.48)**, **Persamaan (2.51)**, dan **Persamaan (2.52)**, maka diperoleh parameter PID yang nilainya ditunjukkan pada **Tabel 3.13**.

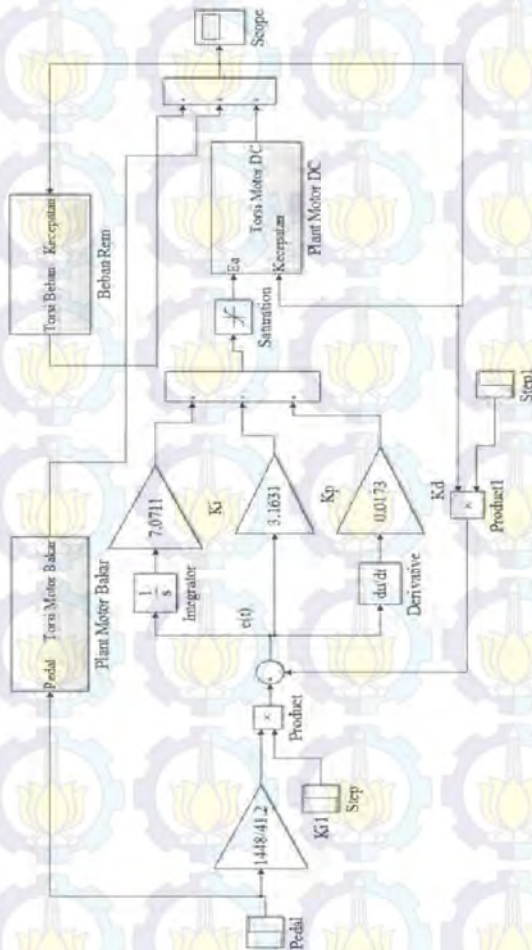
**Tabel 3.13** Nilai Parameter  $K_p$ ,  $K_i$ , dan  $K_d$

Parameter PID		
$K_p$	$K_i$	$K_d$
0,0742	0,2236	0,0009
0,2776	0,3162	0,0029
0,9635	1	0,0078
3,1631	7,0711	0,0173

Nilai Parameter  $K_p$ ,  $K_i$ , dan  $K_d$  yang telah didapatkan kemudian diuji untuk ditentukan nilai parameter PID mana yang paling baik untuk menyelesaikan permasalahan regulator pada sistem HEV.

### 3.7.1 Desain Simulink PID-LQR

Blok Simulink PID-LQR yang digunakan ditunjukkan pada Gambar 3.33.



Gambar 3.33 Desain Simulink PID-LQR





--- Halaman ini sengaja dikosongkan ---

## **BAB 4**

### **PENGUJIAN DAN ANALISIS**

#### **4.1 Gambaran Umum Pengujian Sistem**

Pengujian sistem dilakukan untuk mengetahui performansi kontroler yang sudah dirancang dalam menyelesaikan masalah pada sistem. Pengujian dilakukan dengan melihat pengaruh kinerja motor DC sebagai penggerak pembantu pada sistem *Hybrid Electric Vehicle*. Pada Tugas Akhir ini pengujian sistem dilakukan secara simulasi dan implementasi untuk melihat performansi dari *Hybrid Electric Vehicle* terhadap adanya beban berlebih berupa rem elektromagnetik. Pengujian dilakukan terhadap beberapa keadaan yaitu pada saat ICE diberikan beban tanpa bantuan motor DC dan pada saat ICE dan motor DC menanggung beban bersama menggunakan kontroler PID- *Linear Quadratic Regulator*.

#### **4.2 Simulasi Sistem**

Pada Tugas Akhir ini, simulasi dilakukan dengan menggunakan software MATLAB R2013a untuk mengetahui performansi sistem sebelum dan sesudah diberi kontroler.

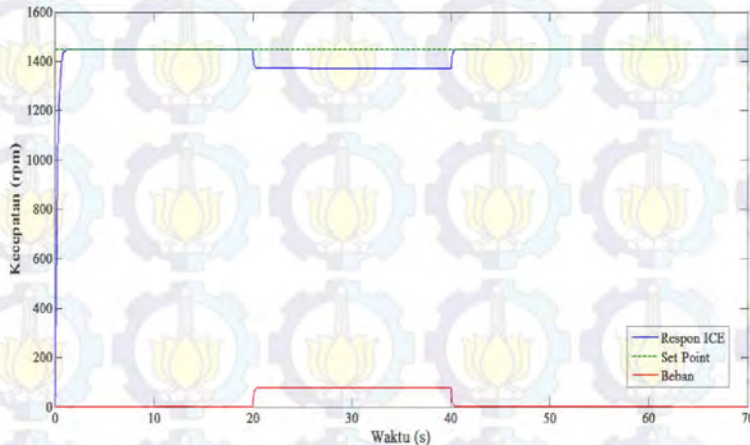
##### **4.2.1 Simulasi ICE Beban Nominal Tanpa Bantuan Motor Listrik**

Simulasi ini bertujuan untuk melihat respon ICE terhadap adanya efek beban. Pada proses pengujian, ICE akan diberi sinyal masukan berupa sinyal *step* sebesar 1448 rpm dengan diberi tiga macam beban transien yang berbeda. Klasifikasi beban untuk permasalahan regulator ini ditentukan dari arus yang diberikan pada rem elektromagnetik. Variasi beban yang diperbolehkan dalam pengujian kontroler harus sesuai dengan pemodelan nominal untuk ICE (d disesuaikan dengan *range* pembebanan nominal), sehingga dibatasi beban yang diperbolehkan adalah pada *range* arus beban 0,56 A– 1,14 A.

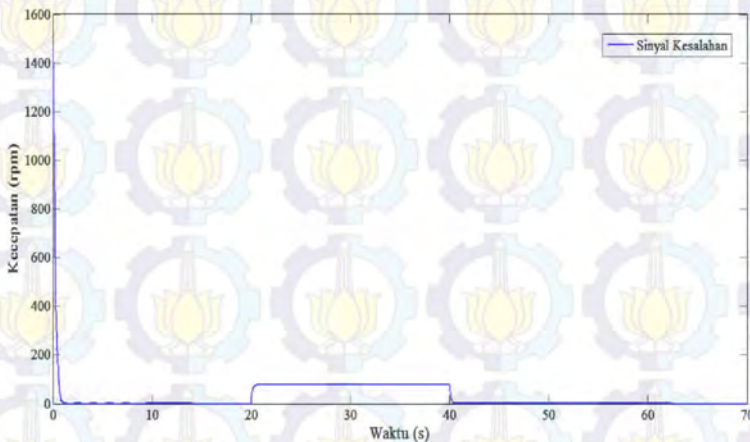
##### **a. Pengujian dengan Arus Beban 0,6 A**

Pada pengujian pertama dilakukan dengan arus rem sebesar 0,6 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 77,4 rpm. Hasil simulasi respon ICE saat diberi beban transien dapat dilihat pada **Gambar 4.1**. Pada **Gambar 4.1** dapat terlihat bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai

*steady state*. Sedangkan, sinyal kesalahan karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE. Sinyal kesalahan sistem dapat dilihat pada **Gambar 4.2**.



**Gambar 4.1** Respon Kecepatan Putar ICE dengan Arus Beban 0,6 A

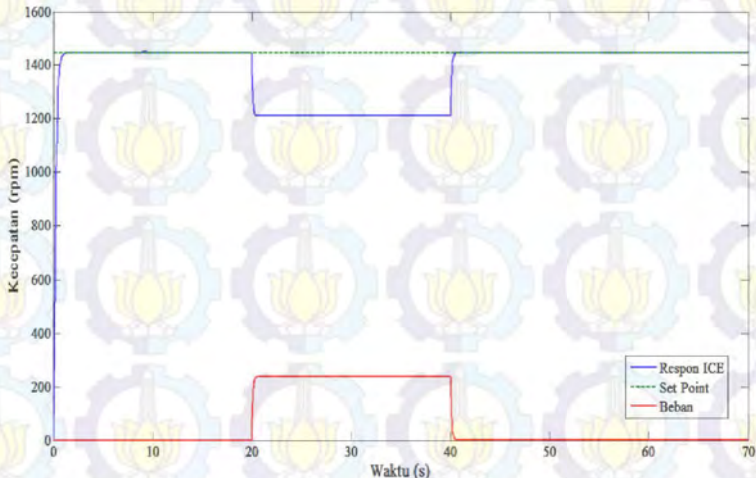


**Gambar 4.2** Sinyal Kesalahan ICE dengan Arus Beban 0,6 A

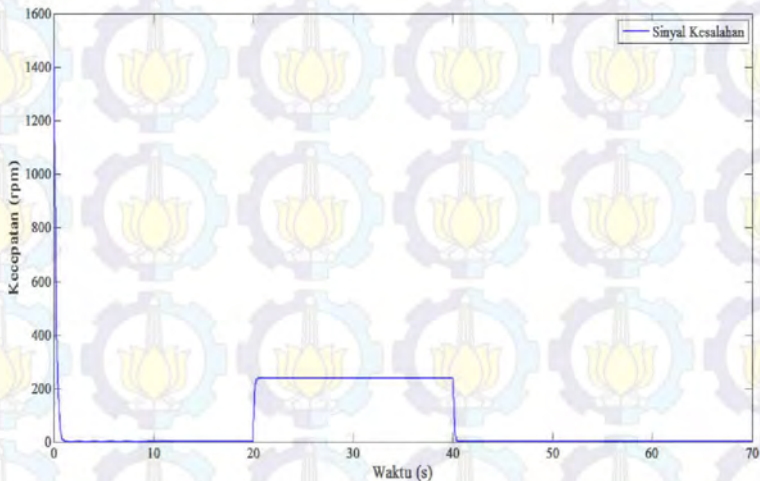
#### b. Pengujian dengan Arus Beban 0,7 A

Pada pengujian ketiga dilakukan dengan memberikan arus rem sebesar 0,7 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 238 rpm. Hasil simulasi terlihat

pada **Gambar 4.3** bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai *steady state*. Sedangkan, sinyal kesalahan karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE. Sinyal kesalahan sistem dapat dilihat pada **Gambar 4.4**.



**Gambar 4.3** Respon Kecepatan Putar ICE dengan Arus Beban 0,7 A

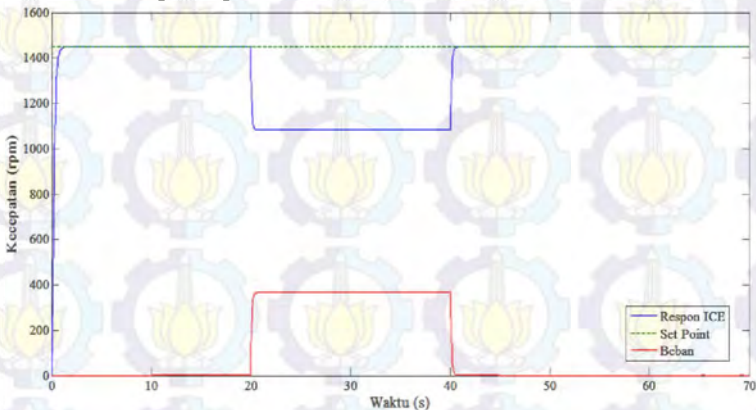


**Gambar 4.4** Sinyal Kesalahan ICE dengan Arus Beban 0,7 A

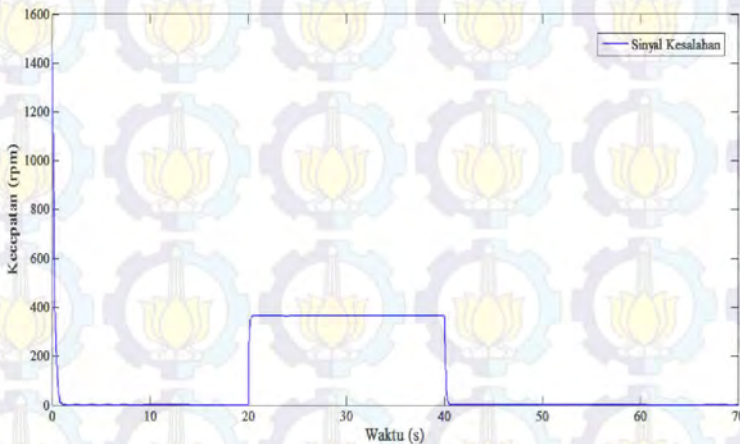


### c. Pengujian dengan Arus Beban 0,8 A

Pada pengujian kedua dilakukan dengan beban berupa arus rem sebesar 0,8 A yang mengakibatkan penurunan kecepatan secara transien terhadap kecepatan putar ICE sebesar 365 rpm. Hasil simulasi ditampilkan pada **Gambar 4.5** bahwa beban diberikan selama selang beberapa waktu, kemudian setelah beban dihilangkan respon ICE kembali mencapai *steady state*. Sedangkan, sinyal kesalahan pada **Gambar 4.6** karena tidak diberikan kontroler, maka sama dengan penurunan kecepatan putar ICE.



**Gambar 4.5** Respon Kecepatan Putar ICE dengan Arus Beban 0,8 A



**Gambar 4.6** Sinyal Kesalahan ICE dengan Arus Beban 0,8 A

Dari ketiga pengujian yang telah dilakukan dengan variasi beban yang berbeda dapat dilihat bahwa pada ketiga pengujian tersebut kesalahan akan bertambah saat terdapat beban dan kembali nol saat beban sudah tidak ada. Hal ini tidak sesuai dengan yang diharapkan, yaitu pada permasalahan regulator, kecepatan motor akan tetap stabil pada kondisi *steady state* mencapai kesalahan nol meskipun ICE diberi beban. Tetapi pada ketiga pengujian yang telah dilakukan, ICE saja tidak mampu digunakan mencapai kesalahan nol saat diberikan beban sehingga perlu diberikan tenaga penggerak pembantu yaitu motor DC.

#### 4.2.2 Simulasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID-Linear Quadratic Regulator

Simulasi ini bertujuan untuk melihat respon ICE dengan bantuan motor DC saat ada beban transien. Beban yang diberikan juga berupa arus rem terdapat 3 macam yaitu 0,6 A, 0,7, dan 0,8 A. Pengujian ini bertujuan untuk menyelesaikan permasalahan regulator dengan menggunakan kontroler PID-Linear Quadratic Regulator untuk mengontrol motor DC agar dapat memberikan bantuan berupa torsi kepada ICE.

##### 4.2.2.1 Model Simulasi

Pemilihan parameter PID untuk mempertahankan kecepatan pada permasalahan regulator berdasarkan nilai kesalahan terkecil dilihat dari nilai IAE (*Integral Absolute Error*) yang terkecil untuk setiap pembebanan. Berdasarkan hasil simulasi (simulink MATLAB dapat dilihat pada **Lampiran D**) didapatkan nilai IAE saat arus beban 0,6 A, 0,7 A, dan 0,8 A pada **Tabel 4.1**, **Tabel 4.2**, dan **Tabel 4.3**. Pada **Tabel 4.1**, **Tabel 4.2**, dan **Tabel 4.3** dapat dilihat bahwa nilai IAE terkecil didapat dengan parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dengan matriks

$$A \begin{bmatrix} - & - \\ - & - \end{bmatrix} B \begin{bmatrix} - \\ - \end{bmatrix}$$

$$R \begin{bmatrix} & \\ & \end{bmatrix} Q \begin{bmatrix} \\ \\ \end{bmatrix} P \begin{bmatrix} \\ \\ \end{bmatrix}$$



sehingga parameter PID tersebut dianggap paling baik untuk dapat membantu kinerja ICE memperbaiki kinerja HEV mencapai *set point* ketika diberi beban.

**Tabel 4.1** Nilai IAE Saat Arus Beban 0,6 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[ ]	[ ]	0,0742	0,2236	0,0009	155,1	
[ ]	[ ]	0,2776	0,3162	0,0029	108,8	
[ ]	[ ]	0,9635	1	0,0078	33,76	
[ ]	[ ]	3,1631	7,0711	0,0173	4,773	

**Tabel 4.2** Nilai IAE Saat Arus Beban 0,7 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[ ]	[ ]	0,0742	0,2236	0,0009	293,4	
[ ]	[ ]	0,2776	0,3162	0,0029	206,6	
[ ]	[ ]	0,9635	1	0,0078	64,69	
[ ]	[ ]	3,1631	7,0711	0,0173	9,149	

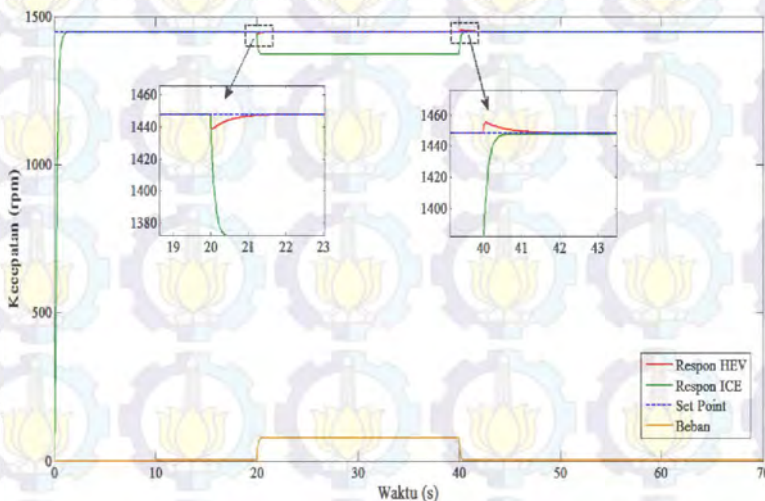
**Tabel 4.3** Nilai IAE Saat Beban 0.8 A

Matriks Pembobot		Parameter PID			IAE	J
R	Q	$K_p$	$K_i$	$K_d$		
[ ]	[ ]	0,0742	0,2236	0,0009	385,4	
[ ]	[ ]	0,2776	0,3162	0,0029	277,1	
[ ]	[ ]	0,9635	1	0,0078	85,25	
[ ]	[ ]	3,1631	7,0711	0,0173	12,06	

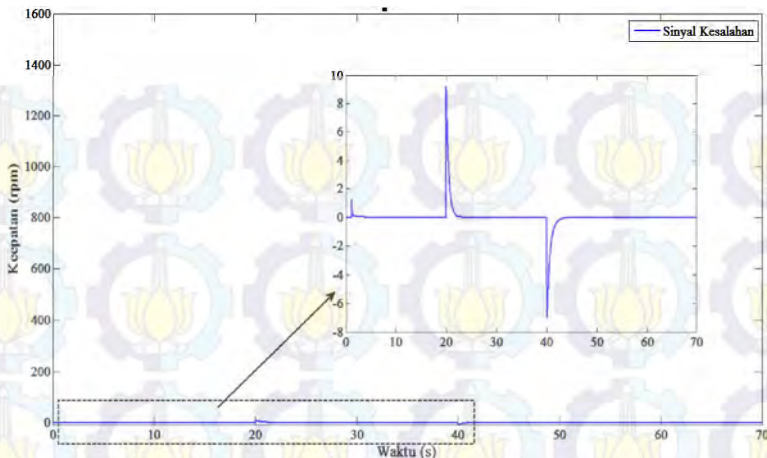
**a. Pengujian dengan Arus Beban 0,6 A**

Pengujian pertama dilakukan dengan memberikan arus beban pada rem elektromagnetik sebesar 0,6 A sehingga terjadi penurunan kecepatan putar sebesar 77,4 rpm. Ketika terjadi penurunan kecepatan, motor DC dengan menggunakan kontroler PID-LQR dapat membantu kinerja ICE untuk memperbaiki performansinya mencapai *set point* dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 9 rpm. seperti yang terlihat pada **Gambar 4.7**. Sinyal kesalahan dari keseluruhan sistem terhadap *set point* ketika terdapat beban seperti yang ditampilkan pada **Gambar 4.8**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Pada **Gambar 4.8** dapat dilihat bahwa motor DC menggunakan kontroler PID-LQR mampu mencapai kesalahan hingga mendekati nol dengan menghilangkan kesalahan sebesar 99,38 % ketika sistem diberikan beban. Sinyal kontrol dapat dilihat pada **Gambar 4.9** dimana sinyal kontrol dikondisikan aktif ketika ICE berada pada kondisi *steady state* yaitu pada detik ke-1,2. Pada Tugas Akhir ini, ICE dianggap telah mencapai kondisi *steady state* dan pemberian beban menyebabkan penurunan kecepatan pada ICE. Ketika ICE telah mencapai *set point*,

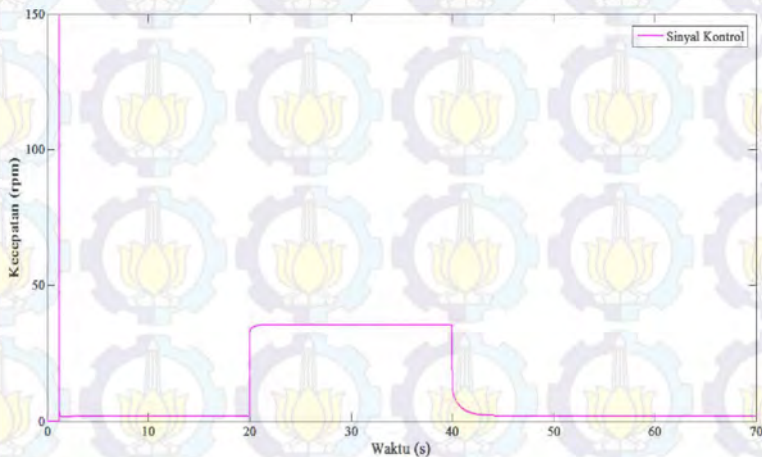
tidak terdapat kesalahan antara sinyal keluaran sistem dengan *set point* sehingga sinyal kontrol tidak bekerja. Ketika sistem diberikan beban pada selang waktu tertentu terjadi penurunan kecepatan sehingga terdapat kesalahan antara sinyal keluaran sistem dengan *set point*. Saat terjadi kesalahan maka sinyal kontrol bekerja untuk menghilangkan kesalahan yang terjadi. Saat beban dilepaskan, ICE kembali lagi pada kondisi *steady state* dan sinyal kontrol tidak lagi bekerja, sehingga sinyal kontrol hanya akan bekerja ketika sistem diberikan beban. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*. Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator.



**Gambar 4.7** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,6



**Gambar 4.8** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,6 A



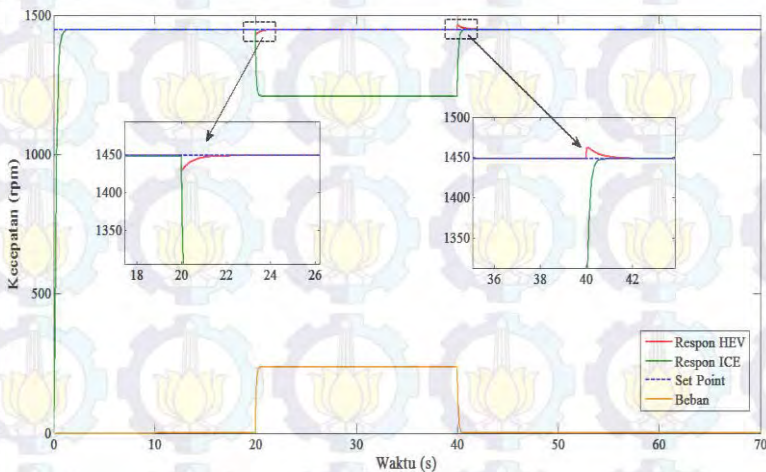
**Gambar 4.9** Sinyal Kontrol Motor DC Saat Arus Beban 0,6 A

#### **b. Pengujian dengan Arus Beban 0,7 A**

Pengujian kedua dilakukan dengan memberikan arus rem yang lebih tinggi yaitu sebesar 0,7 A. Karena diberikan beban yang lebih tinggi, maka penurunan kecepatan juga lebih tinggi yaitu sebesar 238 rpm. Pada **Gambar 4.10** dapat terlihat, dengan kontroler PID-LQR, ICE dan motor DC mampu menanggung beban bersama untuk

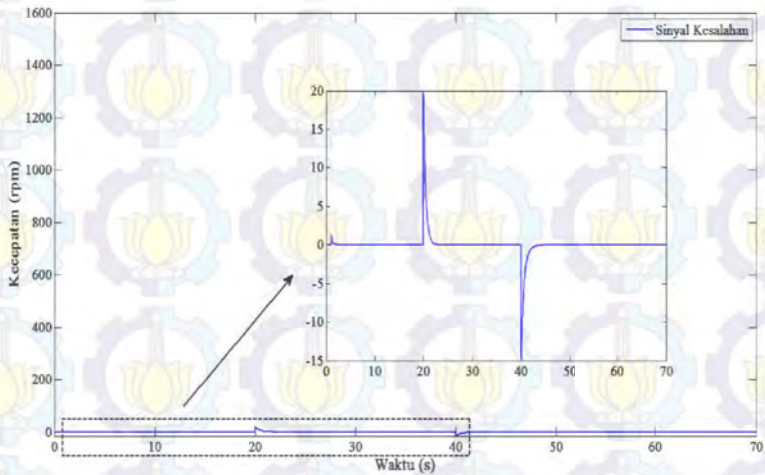


mempertahankan kecepatan menuju nilai *steady state* dari ICE dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 20 rpm. Sinyal kesalahan dari keseluruhan sistem ditampilkan pada **Gambar 4.11**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Motor DC menggunakan kontroler PID-LQR mampu menghilangkan kesalahan sebesar 99,69 % ketika terjadi pembebanan. Hal ini menandakan bahwa kontroler PID-LQR mampu bekerja dengan baik sehingga mampu mencapai kesalahan hingga mendekati nol.. Sinyal kontrol dapat dilihat pada **Gambar 4.12**. Pada Tugas Akhir ini, ICE dianggap telah mencapai kondisi *steady state* dan pemberian beban menyebabkan penurunan kecepatan pada ICE. Ketika ICE telah mencapai *set point*, sinyal kontrol tidak bekerja, artinya tidak terdapat kesalahan antara sinyal keluaran sistem dengan *set point*. Ketika sistem diberikan beban pada selang waktu tertentu, sinyal kontrol bekerja hingga beban dilepaskan. Saat beban dilepaskan, ICE kembali lagi pada kondisi *steady state* dan sinyal kontrol tidak lagi bekerja. Sehingga sinyal kontrol hanya akan bekerja ketika sistem diberikan beban. Pada pemberian beban yang lebih tinggi sinyal kontrol yang bekerja juga akan lebih besar untuk mengurangi kesalahan sistem.

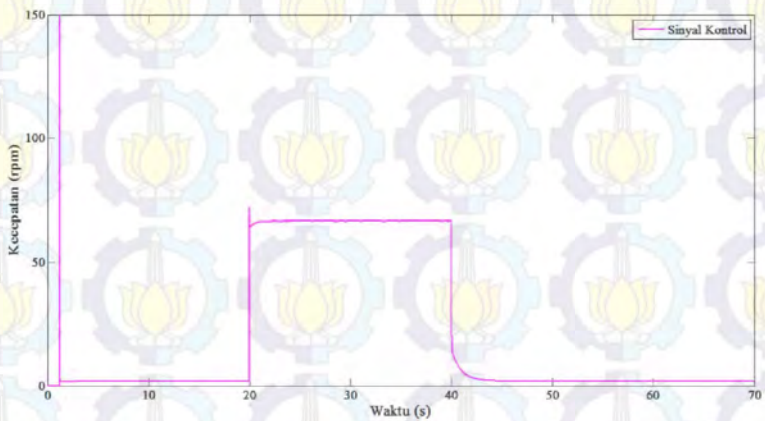


**Gambar 4.10** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,7 A

Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*.



**Gambar 4.11** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,7 A

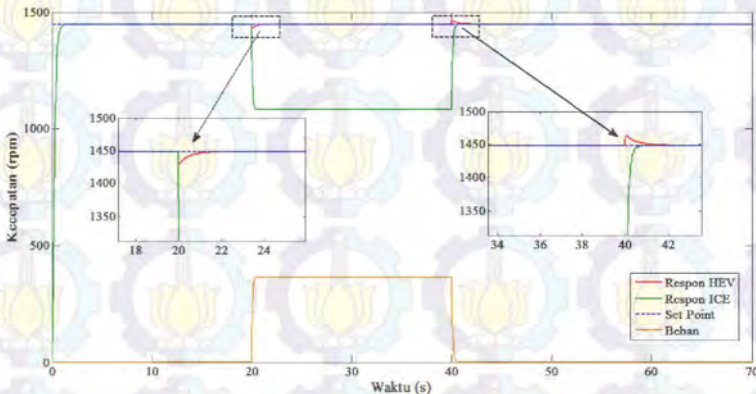


**Gambar 4.12** Sinyal Kontrol Motor DC Saat Arus Beban 0,7 A

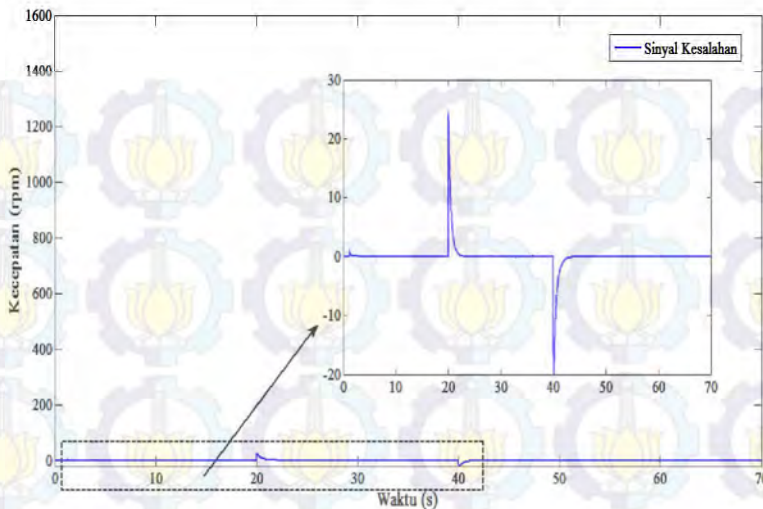


### c. Pengujian dengan Arus Beban 0,8 A

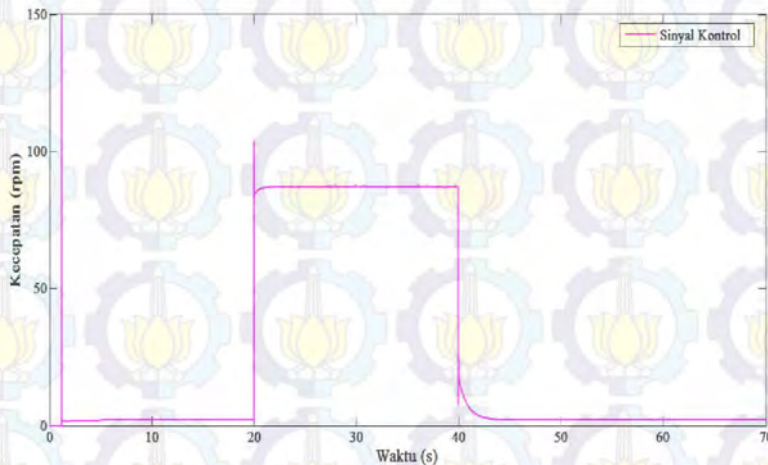
Pengujian kedua dilakukan dengan memberikan arus yang lebih tinggi pada rem elektromagnetik sebesar 0,8 A sehingga terjadi penurunan kecepatan putar sebesar 365 rpm. Dengan kontroler PID-LQR, ICE dan motor DC mampu menanggung beban bersama untuk mempertahankan kecepatan menuju nilai *steady state* dari ICE seperti yang terlihat pada **Gambar 4.13** dalam waktu 0,7 detik pada titik awal sistem diberikan beban dan saat beban dihilangkan dengan *undershoot* dan *overshoot* sebesar 25 rpm. Sinyal kesalahan dari keseluruhan sistem terhadap *set point* ketika terdapat beban seperti yang ditampilkan pada **Gambar 4.14**. Saat beban dihilangkan secara transien, motor DC perlahan-lahan melambat hingga ICE kembali tak berbeban dan kecepatan putar kembali menuju nilai *steady state*. Motor DC menggunakan kontroler PID-LQR mampu menghilangkan kesalahan sebesar 99,80 %. Sinyal kontrol dapat dilihat pada **Gambar 4.15**. Pada pemberian beban yang lebih tinggi lagi sinyal kontrol yang bekerja juga akan lebih besar untuk mengurangi kesalahan sistem. Sinyal kontrol dikondisikan ke dalam 5 V sesuai dengan akuisisi data mikrokontroler sehingga sinyal kontrol dapat menyesuaikan suplai tegangan *driver*. Sinyal kontrol yang bekerja diberikan batasan atau saturasi sinyal kontrol yang setara dengan tegangan suplai motor DC sebesar 150 VDC. Jika tidak diberikan batasan atau saturasi sinyal kontrol, maka ketika melebihi tegangan yang diperbolehkan akan terjadi kerusakan pada aktuator.



**Gambar 4.13** Respon Kecepatan Putar HEV dengan Kontroler PID-LQR Saat Arus Beban 0,8 A



**Gambar 4.14** Sinyal Kesalahan dari Sistem Saat Arus Beban 0,8 A



**Gambar 4.15** Sinyal Kontrol Motor DC Saat Arus Beban 0,8 A

Dari ketiga pengujian yang telah dilakukan dengan variasi beban yang berbeda, motor DC dengan menggunakan kontroler PID-LQR dapat membantu kinerja ICE untuk memperbaiki performansinya mencapai *set point* saat ada beban. Salah satu tugas kontroler adalah

meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal *set point* dan sinyal aktual. Berdasarkan hasil pengujian secara simulasi, perancangan kontroler PID-LQR untuk mengatur motor DC dalam membantu ICE sudah sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang senantiasa (diinginkan) sama dengan sinyal *set point*.

Jika dilihat dari analisis kestabilannya, *eigen value* dari ketiga pengujian ini memiliki nilai kurang dari nol atau berada pada sebelah kiri sumbu imajiner sehingga dapat disimpulkan bahwa respon stabil. Hal ini menandakan bahwa kontroler PID-LQR memiliki performansi yang baik untuk permasalahan regulasi sehingga dapat mengatur motor DC untuk membantu kinerja ICE mempertahankan kondisi *steady state* nya.

### 4.3 Implementasi Sistem

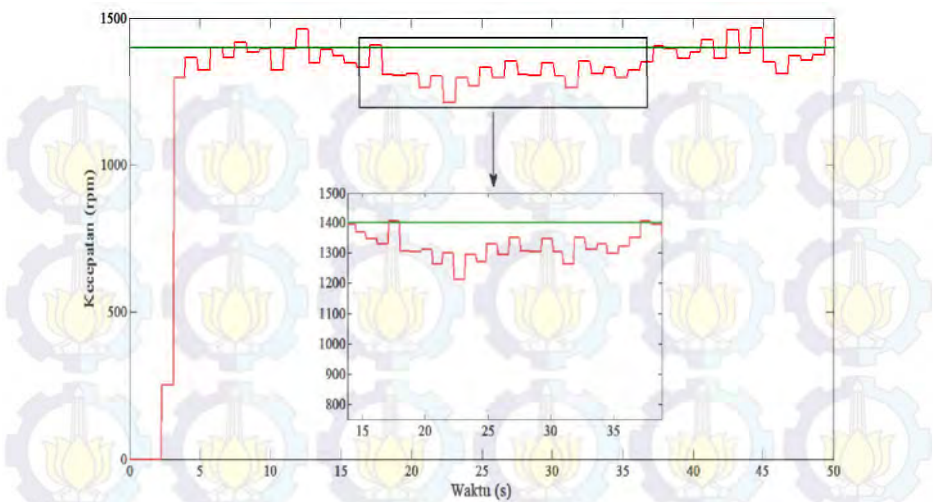
Implementasi sistem pengaturan kecepatan HEV dilakukan pada *plant* secara langsung. Pengujian dilakukan dengan menghubungkan *plant* HEV terhadap kontroler yang telah dirancang pada MATLAB R2013a melalui Arduino *board* dan *driver* motor DC. Pembacaan *rotary encoder* dan pedal masuk pada *analog read* Arduino dan menjadi *set point*. Sinyal kontrol dari MATLAB dikirim melalui pengiriman data serial ke *driver* motor DC untuk menggerakkan motor DC sesuai pengaturan PWM nya. Pengujian pada sistem dilakukan dengan memberikan variasi beban berupa tegangan rem nominal sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC.

#### 4.3.1 Implementasi ICE Beban Nominal Tanpa Bantuan Motor Listrik

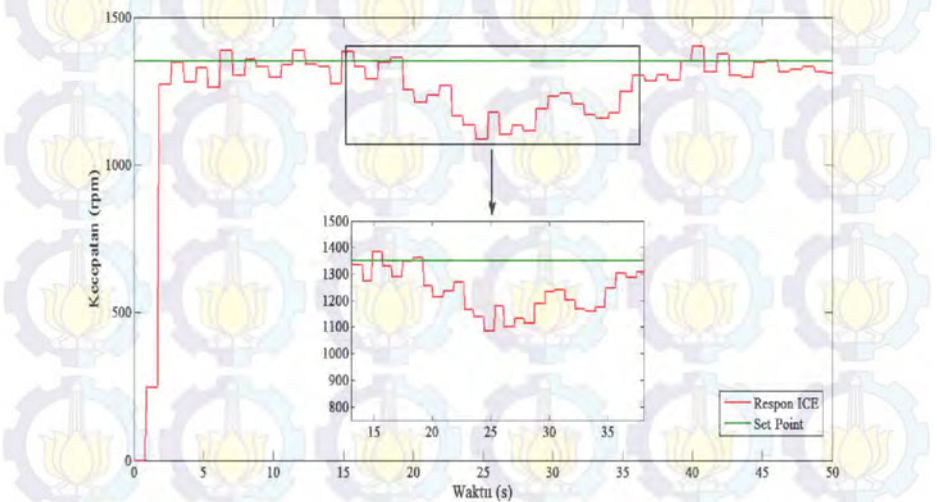
Pada pengujian ini, hanya dilihat respon ICE terhadap adanya pemberian tegangan sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC pada rem elektromagnetik yang dapat menyebabkan penurunan kecepatan pada ICE. Respon ICE dapat dilihat pada **Gambar 4.16** dan **Gambar 4.17**.

Pada **Gambar 4.16** dan **Gambar 4.17** dapat dilihat bahwa pada saat rem elektromagnetik diberi tegangan sebesar 50 VDC, dan 57,15 VDC, maka kecepatan ICE akan menurun. Semakin besar tegangan yang diberikan pada rem elektromagnetik, maka semakin besar pula penurunan kecepatan yang terjadi pada ICE.





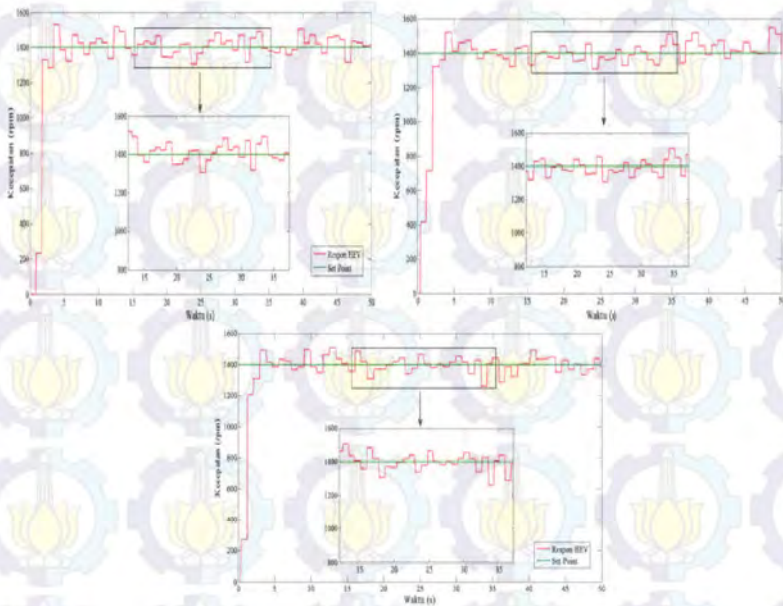
**Gambar 4.16** Respon Kecepatan Putar ICE Hasil Impementasi dengan Tegangan Rem 50 VDC



**Gambar 4.17** Respon Kecepatan Putar ICE Hasil Implementasi dengan Tegangan Rem 57,15 VDC

#### 4.3.2 Implementasi ICE Beban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler PID-Linear Quadratic Regulator

Pada pengujian ini, ICE sebagai penggerak utama dibantu oleh motor DC sebagai penggerak pembantu. Setelah itu diberikan tegangan pada rem elektromagnetik sebesar 42,86 VDC, 50 VDC, dan 57,15 VDC sebagai efek beban yang dapat menyebabkan penurunan kecepatan pada ICE. Respon HEV secara keseluruhan dapat dilihat pada **Gambar 4.18**.



**Gambar 4.18** Respon Kecepatan Putar HEV Hasil Implementasi Menggunakan Kontroler PID-LQR dengan 3 Variasi Beban

Pada **Gambar 4.18** dapat dilihat bahwa pada pemberian 3 variasi beban berupa tegangan rem, motor DC menggunakan kontroler PID-LQR mampu membantu mesin bakar untuk mengembalikan kecepatan hingga mendekati *set point* pada saat terjadi pembebanan meskipun masih terdapat  $e_{ss}$  sebesar 20,67 % ketika pemberian tegangan rem 42,86 VDC,  $e_{ss}$  sebesar 28,8 % ketika pemberian tegangan rem 50 VDC, dan  $e_{ss}$  sebesar 26,6 % ketika pemberian tegangan rem 57,15 VDC.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis, didapatkan kesimpulan sebagai berikut :

1. Parameter kontroler PID berupa  $K_p$ ,  $K_i$  dan  $K_d$  yang optimal dapat dicari menggunakan pendekatan *Linear Quadratic Regulator* berdasarkan penerapan aturan tertentu yang sesuai.
2. Nilai parameter  $K_p$ ,  $K_i$ , dan  $K_d$  merupakan nilai parameter PID terbaik untuk dapat membantu kinerja ICE memperbaiki performansinya mencapai *set point* ketika diberi beban.
3. Hasil pengujian secara simulasi menggunakan kontroler PID-LQR, motor DC mampu membantu kerja ICE pada saat terdapat beban sehingga kecepatan yang diinginkan dapat dipertahankan sehingga dapat menghilangkan kesalahan sebesar 99,38 % ketika sistem diberikan arus beban 0,6 A, 99,69 % ketika sistem diberikan arus beban 0,7 A, dan 99,80 % saat sistem diberikan arus beban sebesar 0,8 A.
4. Pada hasil implementasi, motor DC juga mampu membantu ICE sehingga kecepatan putar HEV mendekati *set point*, meskipun masih terdapat *error steady state* hingga 28,8 %.

#### 5.2 Saran

Pada Tugas Akhir ini, pemilihan matriks **Q** dan **R** untuk memperoleh parameter PID masih menggunakan metode *trial and error*. Diharapkan pada penelitian selanjutnya, pemilihan matriks **Q** dan **R** dapat menggunakan metode komputasi cerdas agar mendapatkan hasil yang lebih optimal. Selain itu, identifikasi dan pemahaman tentang *plant* yang digunakan dapat membantu perancangan sistem yang lebih baik.





## DAFTAR LAMPIRAN

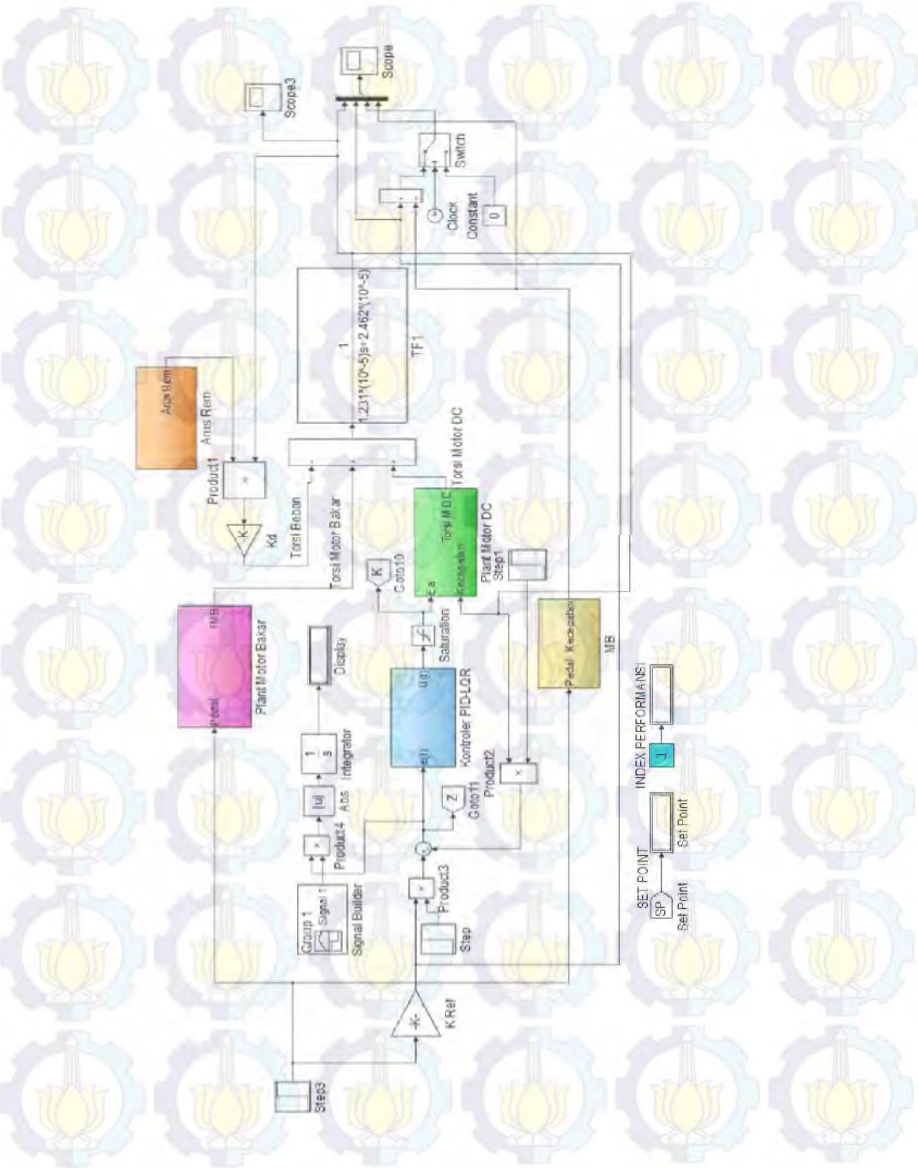
<b>Lampiran A</b>	Blok LabView untuk Identifikasi Statis .....	107
<b>Lampiran B</b>	Diagram Simulink Sistem HEV .....	108
<b>Lampiran C</b>	Diagram Simulink <i>Plant</i> Motor DC .....	109
<b>Lampiran D</b>	Diagram Simulink Pencarian Nilai IAE.....	110
<b>Lampiran E</b>	Diagram Simulink Implementasi Sistem HEV .....	111
<b>Lampiran F</b>	Program MATLAB .....	112
<b>Lampiran G</b>	Program Arduino untuk Pembacaan Kecepatan Motor DC .....	113
<b>Lampiran H</b>	Program Arduino untuk Implementasi .....	114
<b>Lampiran I</b>	Program Perhitungan RMSE.....	115



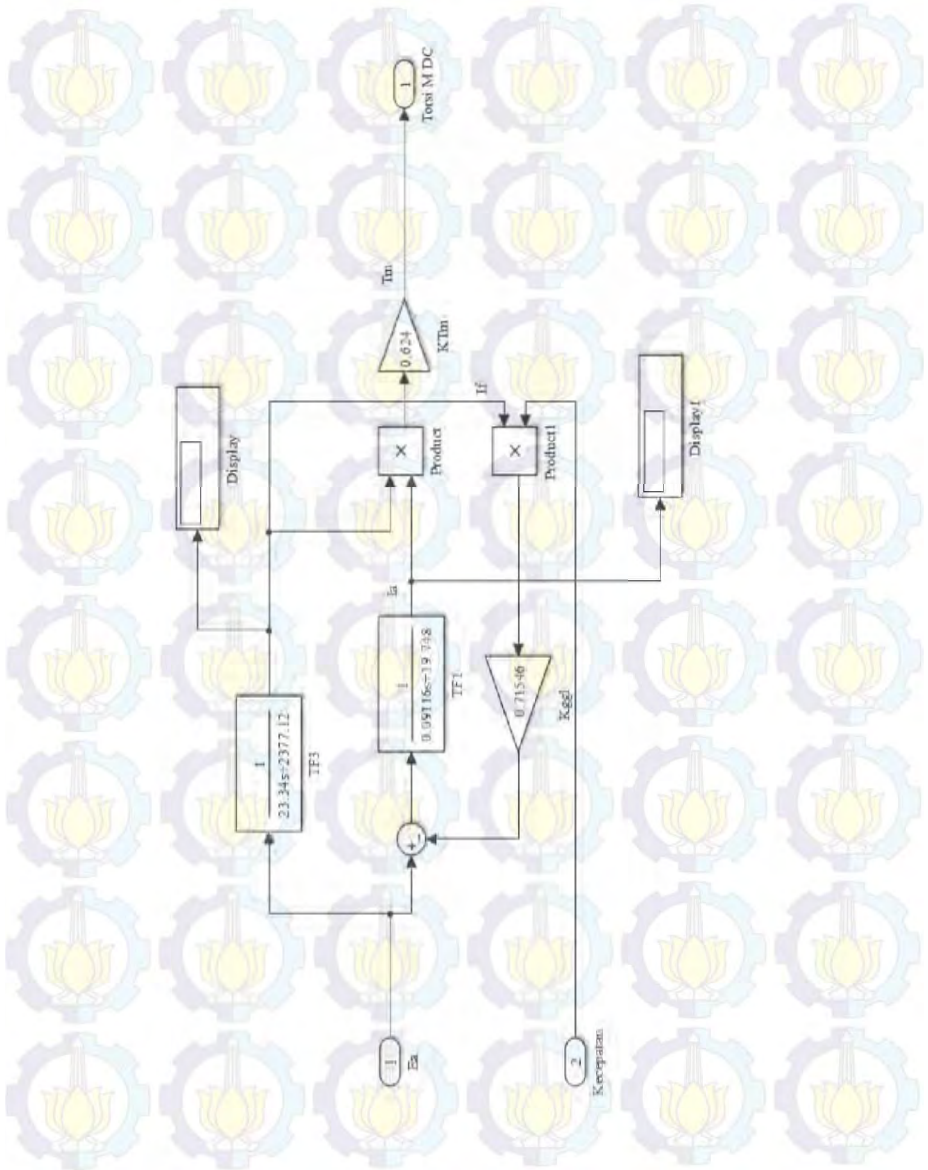




**B. Diagram Simulink Sistem HEV**

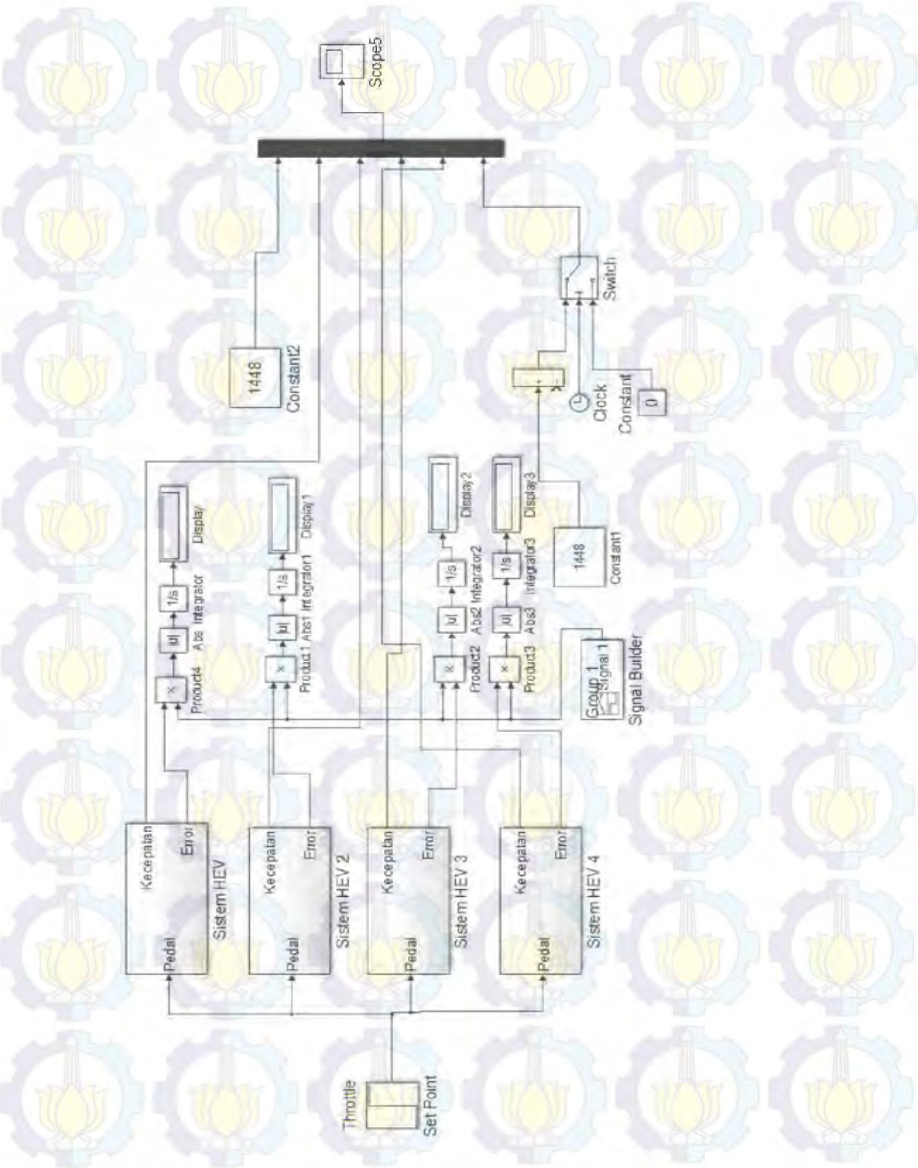


### C. Diagram Simulink *Plant* Motor DC

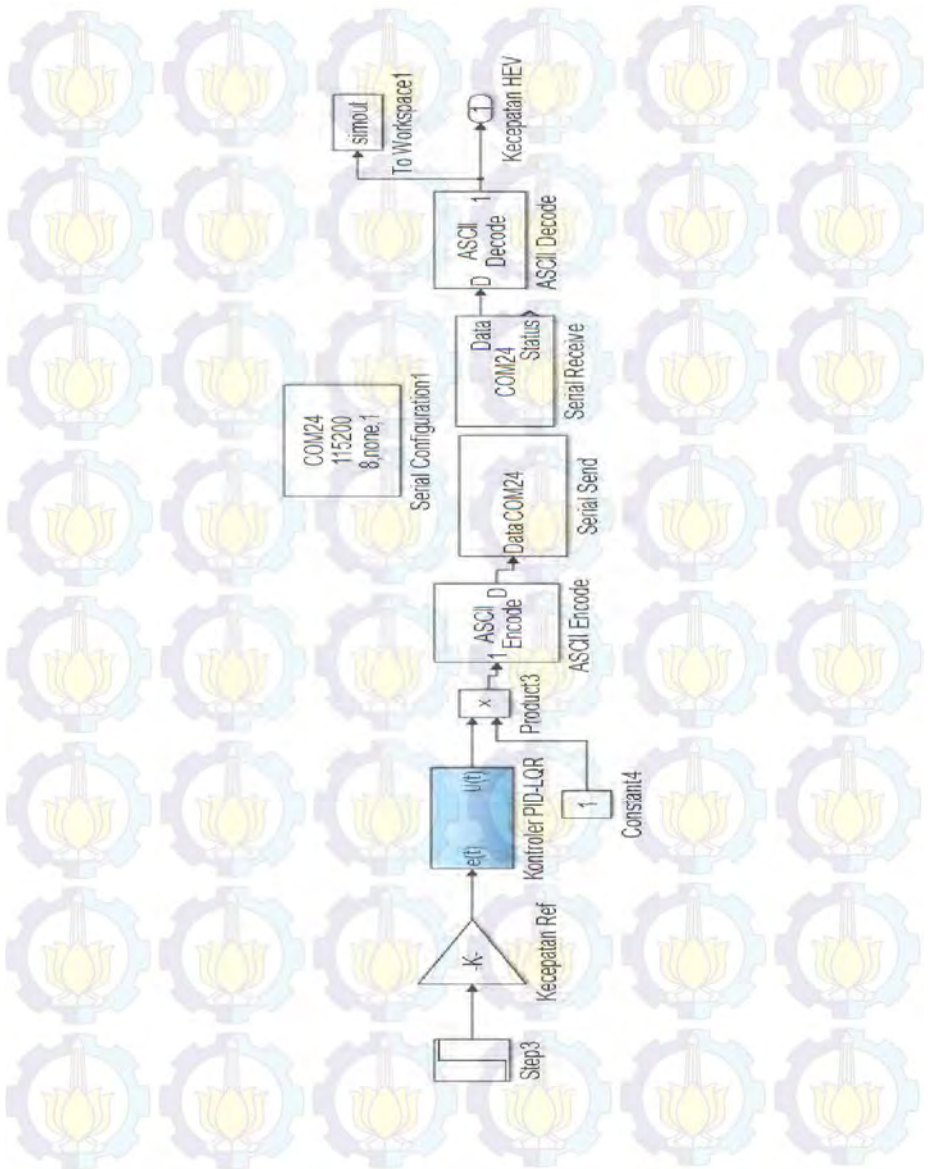




**D. Diagram Simulink Pencarian nilai IAE**



### E. Diagram Simulink Implementasi Sistem HEV



## F. Program MATLAB

```
% Judul Tugas Akhir :  
% Pengaturan Kecepatan pada Simulator Paralel Hybrid Electric Vehicle  
% Menggunakan Metode PID-Linear Quadratic Regulator  
%  
% Fannieshe Hamada  
% 2211100207  
%  
% Jurusan Teknik Elektro ITS  
% Surabaya 2015  
  
% Parameter PID-LQR  
A = [0 1 0; 0 0 1; 0 -635.9768 -76.4785]; % Matriks A  
B = [0; 0; -12295.58]; % Matriks B  
Q = [50 0 0; 0 1 0; 0 0 0]; % Matriks Pembobot Q  
R = [1]; % Matriks Pembobot R  
  
% Perhitungan parameter Ki, Kp, dan Kd  
[K,P,EV]= lqr(A,B,Q,R)  
% P = Riccati Matrix  
% EV = Eigen Value
```

## G. Program Arduino untuk Pembacaan Kecepatan Motor DC

```
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(encoder, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    delay(1);
    if (Serial.read() == 'M')
    {
      while (Serial.available())
      {
        Serial.read();
      }
      unsigned long waktu = pulseIn(encoder, HIGH);
      float jeda = waktu;
      jeda = jeda / 1000000;
      float kecepatan = (1 / (jeda * 72)) * 60;
      float y = 229.9536927+(0.680455391*kecepatan);
      word rpm = y;
      Serial.print(rpm);
    }
  }
}
```



## H. Program Arduino Untuk Implementasi

```
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;
char disp[1];
int sinyalkontrol;
int sinyalkontrolarduino;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(encoder, INPUT);
    pinMode(13, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    if (Serial.available()>0)
    { Serial.parseInt();}
    delay(1);
    unsigned long waktu = pulseIn(encoder, HIGH);
    float jeda = waktu;
    jeda = jeda / 1000000;
    float kecepatan = (1 / (jeda * 72)) * 60;
    float y = 229.9536927+(0.680455391*kecepatan);
    word rpm = y;
    sprintf(disp, "%4d",rpm);
    Serial.print(disp);

    if (Serial.available()>0){
        sinyalkontrol = Serial.parseInt();
        sinyalkontrolarduino = map(sinyalkontrol,0,150,0,1000);
        digitalWrite(13,LOW);
```

```
delay(sinyalkontrolarduino);  
digitalWrite(13,HIGH);  
delay(1000-sinyalkontrolarduino);  
}  
}
```

## **I. Program Perhitungan RMSE**

```
clc;  
n=size(xxx,1);  
A=xxx(:,2);  
M=xxx(:,3);  
sumei=0;  
for i=1:n;  
    ei=(A(i)-M(i))/A(i)  
    sumei=sumei+ei^2  
end  
sumei  
n  
RMSE=sqrt(sumei/n)
```





## DAFTAR PUSTAKA

- [1] M. Mohebbi, M. Charkhgard and M. Farrokhi, "Optimal Neuro-Fuzzy Control of Parallel Hybrid Electric Vehicles", *Vehicle Power and Propulsion, IEEE Conference*, pp.26-30, 7-9 Sept, 2005
- [2] W. A. Candra, Pengaturan Kecepatan pada Parallel Hybrid Electric Vehicle Menggunakan Linear Quadratic Regulator Berdasarkan Particle Swarm Optimization, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [3] G. Satrio Aji Wibowo, Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle, Surabaya: Jurusan Teknik Elektro ITS, 2010.
- [4] W. Sriwidodo, Perancangan dan Implementasi Pengendalian Traksi Berbasis Neuro Fuzzy untuk Simulator Kendaraan Hybrid pada Mode Akselerasi, Surabaya: Jurusan Teknik Elektro ITS, 2010.
- [5] Y. Du, J. Gao, L. Yu, J. Song, F. Zhao and W. Zhan, "HEV System Based on Electric Variable Transmission," in *Vehicle Power and Propulsion Conference IEEE*, 2009.
- [6] A. Thonthowi, Pengaturan Kecepatan pada Parallel Hybrid Electric Vehicle Menggunakan Self Tuning Proportional Integral Derivative Berdasarkan Genetic Algorithm, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [7] A. Rahadi, "Cara Kerja Mesin 2 Tak dan 4 Tak," Juli 2014.
- [8] A. Endarwati, Perancangan Sistem Pengaturan Kecepatan pada Simulator Parallel Hybrid Electric Vehicle (PHEV) Menggunakan Metode State Dependent-Linear Quadratic Regulator, Surabaya: Jurusan Teknik Elektro ITS, 2014.
- [9] D. SDP, "Drive Your System With Timing Belt," Juli 2013.
- [10] R. Hardiningrat, Pembuatan Alat Ukur Kedalaman Lubang Sumur Bor, Bandung: Teknik Mesin Universitas Pasundan, 2013.
- [11] "Index of Arduino Board," November 2013. [Online]. Available: <http://www.arduino.cc/en/Main/arduinoBoardUno>. [Accessed Mei 2015].
- [12] I. Pavel Jakoubek, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods," in *Konference Studenske Tvurci-Cinnosti*, 2009.

- [13] K. Ogata, Modern Control Engineering, Prentice-Hall Inc, 1970.
- [14] N. Gamayanti, Diktat Kuliah Dasar Sistem Pengaturan, Surabaya: ITS, 2012.
- [15] D. S. Naidu, Optimal Control System, New York: CRC Press, 2003.
- [16] F. L. Lewis and V. L. Syrmos, Optimal Control, Inc: CRC Press, 1995.
- [17] G. Y. Ruey and R. C. Hwang, "Optimal PID Speed Control of Brushless DC Motors Using LQR Approach," in *IEEE International Conference on Systems, Man and Cybernetics*, Taiwan, 2004.
- [18] J. B. He, Q. G. Wang and T. H. Lee, "PI/PID Controller Tuning Via LQR Approach," *Proceedings of The 37th IEEE Conference on Decision and Control*, vol. 1, pp. 1177-1182, 2005.

## RIWAYAT HIDUP



Penulis bernama lengkap Fanniesha Hamada, Lahir di kota Jakarta pada tanggal 13 Januari 1994 dan merupakan anak kedua dari 3 bersaudara dari pasangan Arifin dan Aryani Nilasari. Setelah lulus dari SMAN 2 Kota Tangerang Selatan pada tahun 2011, penulis melanjutkan studinya di Institut Teknologi Sepuluh Nopember Surabaya (ITS), Jurusan Teknik Elektro dan mengambil bidang studi Teknk Sistem Pengaturan. Pada tahun kedua dan ketiga perkuliahan, penulis aktif dalam bidang organisasi dan kepanitiaan *event* di lingkungan kampus maupun di luar kampus. Pada bulan Juni 2015, penulis mengikuti seminar dan ujian Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro.